**从圆周率到歌曲：对流行音乐生成的一个音乐的仿真网络**

SONG FROM PI: A MUSICALLY P LAUSIBLE NETWORKF OR POP MUSIC GENERATION

杭楚，Raquel Urtasun，Sanja Fidler

计算机科学系

多伦多大学

加拿大 安大略湖 M5S 3G4

{ chuhang1122，urtasun，费德勒} @cs.toronto.edu

**摘要（ABSTRACT）**

We present a novel framework for generating pop music. Our model is a hierarchical Recurrent Neural Network, where the layers and the structure of the hierarchy encode our prior knowledge about how pop music is composed. In particular, the bottom layers generate the melody, while the higher levels produce the drums and chords. We conduct several human studies that show strong preference of our generated music over that produced by the recent method by Google. We additionally show two applications of our framework: **neural dancing** and karaoke, as well as **neural story singing**.

我们提出了一个产生流行音乐的新框架。我们的模型是一个有层次的递归神经网络（RNN），其中层次和层次的结构编码使用了我们的先进知识，这个知识是关于如何组成流行音乐的。特别是，底层产生旋律，而更高层产生鼓点和和弦。我们进行了几项人类研究，这些研究表明，我们的生成音乐比谷歌最近的方法产生的更强。此外，我们还展示了两个框架的应用：**神经舞蹈**和卡拉OK，以及**神经故事唱歌**。

**简介（INTRODUCTION）**

Neural networks have revolutionized many fields. They have not only proven to be powerful in performing perception tasks such as image classification and language understanding, but have also shown to be surprisingly good “artists”. In Gatys et al. (2015), photos were turned into paintings by exploiting particular drawing styles such as Van Gogh’s, Kiros et al. (2015) produced stories about images biased by writing style (e.g., romance books), Karpathy et al. (2016) wrote Shakespeare inspired novels, and Simo-Serra et al. (2015) gave fashion advice.

神经网络使许多领域发生了革命性的变化。他们不仅被证明具有能力执行感知任务，如图像分类和语言理解，也显示出了出众的好“艺术家”（能力）。gatys等人（2015），照片变成了绘画是利用特定的绘画风格，如梵高；Kiros等人（2015）产生的写作风格偏向图片故事（如浪漫书籍）；karpathy等人（2016）受莎士比亚的启发，写小说；Simo Serra等人（2015）给出时尚建议。

Music composition is another artistic domain where neural based approaches have been proposed. Early approaches exploiting Recurrent Neural Networks (Bharucha & Todd (1989); Mozer (1996); Chen & Miikkulainen (2001); Eck & Schmidhuber (2002)) date back to the 80’s.

The main variations between the different models is the representation of the notes and the outputs they produced,which typically encode melody and chord. Most of these approaches were single track, in that they produced only one note **per time step**. The exception is Boulanger-lewandowski et al. (2012) which generated polyphonic music, i.e., simultaneous independent melodies.

音乐创作是另一个以神经为基础的方法的艺术领域。早期的方法利用递归神经网络（Bharucha & Todd (1989);Mozer（1996）;Chen & Miikkulainen (2001); Eck & Schmidhuber (2002)）追溯到80年代。不同模型之间的主要变化是它们所产生的音符和输出的代表性，它们通常用来编码旋律和和弦。这些方法大多是单轨的，因为他们**每次**只生产一个音符。唯一的例外是Boulanger lewandowski等人（2012）产生复调音乐，即同时发生又相互独立的旋律。

In this paper, we aim to generate pop music, where the melody but also chords and other instruments make up what is typically called a song. We draw inspiration from the Song from π by **Macdonald1[[1]](#footnote-1)** a piano video on Youtube, where the pleasing music is created from a sequence of digits of π. This video shows both the randomness and the regularity of music. On one hand, since any possible digit sequence is a subset of the π digit sequence, this implies that pleasing music can be created even from a totally random base signal. On the other hand, the composer uses specific rules such as A **Harmonic Minor scale** and harmonies to convert the digit sequence into a music sheet. It is these rules that play the key role in converting randomness into music.

在本文中，我们的目标是产生流行音乐，就是由旋律、和弦和其他乐器组成的，通常称为歌曲的东西。我们的灵感来自**麦克唐纳德**在YouTube上的一段钢琴视频，视频中是他从π中得到的歌曲，这悦耳的音乐是从一个π的数字序列创建的。这部影片展示了音乐的随机性和规律性。一方面，因为任何可能的数字序列都是π的数字序列的一个子集，这意味着悦耳的音乐甚至可以从一个完全随机的基带信号中产生。另一方面，作曲家使用特定的规则，如**调和小音阶**和和声，将数字序列转换成乐谱。正是这些规则在将随机的东西转换为音乐中发挥了关键作用。

Following the ideas of Songs from π, we aim to generate both the melody as well as accompanying effects such as chords and drums. Arguably, these turn even a not particularly pleasing melody into a well sounding song. We propose a hierarchical approach, where each level is a Recurrent Neural Network producing a key aspect of the song. The bottom layers generate the melody, while the higher levels produce drums and chords. This enables the drum and chord layers to compensate for the melody in order to produce **appleasing music**. Adopting the key idea from Songs from π, we condition our model on the scale type allowing the melody generator to learn the notes that are typically played in a particular scale（音阶）.

根据从π中获得的歌曲的思想，我们致力于创造既有旋律又附带如和弦和鼓效果的音乐。可以说，这些歌曲甚至不是特别悦耳的旋律，而是一首动听的歌曲。我们提出了一种分层的方法，每一层都是一个递归神经网络，产生了歌曲的一个关键方面。底层产生旋律，而更高的层次产生鼓和和弦。为了生产**appleasing music**，这使鼓与弦层为音乐旋律做出补偿。采用来自π的音乐的关键想法，我们对应音阶类型调整我们的模型，让模型发声器去学习那些通常谱在特殊音阶地方的音符。

We train our model on 100 hours of midi music containing user-composed pop songs and video game music. We conduct human studies with music generated with our approach and compare it against a recent approach by Google, showing that our songs are strongly preferred over the baseline. In our human study we also perform an **ablation analysis** of our model. We additionally show two new applications: neural dancing and karaoke as well as neural music singing. As part of the first application we generate a stickman dancing to our music and lyrics that can be sung with, while in the second application we condition on the output of Kiros et al. (2015) which writes a story about an image and convert it into a pop song. We refer the reader to <http://www.cs.toronto.edu/songfrompi/> for our demos and results.

我们在100小时的MIDI音乐中训练我们的模型，其中包含用户创作的流行歌曲和电子游戏音乐。我们用我们的方法产生的音乐进行人类研究，并与谷歌最近的方法进行比较，结果显示我们的歌曲比基线更受欢迎。在我们的人类研究中，我们也对我们的模型进行**烧蚀分析**。我们还展示了两个新的应用：神经舞蹈和像神经音乐唱歌的卡拉OK。作为第一个应用程序的一部分，我们产生了一个火柴人，它在我们可以唱的那部分音乐和歌词中；而在第二应用，

我们把Kiros等人由一幅图像写出的故事作为输出，来把它变成一首流行音乐。我们推荐读者到<http://www.cs.toronto.edu/songfrompi/>看演示和结果。

**2.相关的工作（RELATED WORK）**

Generating music has been an active research area for decades. It brings together machines learn-ing researchers that aim to capture the complex structure of music (Eck & Schmidhuber (2002);Boulanger-lewandowski et al. (2012)), as well as music professionals (Chan et al. (2006)) and enthusiasts (Johnson; Sun) that want to see how far a computer can get to be a real composer. Real-time music generation is also explored for gaming (Engels et al. (2015)).

几十年来，音乐创作一直是一个活跃的研究领域。它汇集了机器学习的研究人员，旨在捕捉音乐的复杂结构（Eck和Schmidhuber（2002）；Boulanger lewandowski等人（2012），以及音乐专业人士（Chan等人，2006），和热心者（Johnson，Sun），他们想看看计算机能成为多么真实的作曲家。实时音乐生成还被探索应用于游戏（恩格斯等人，2015）。

Early approaches mostly instilled knowledge from music theory into generation, by using rules of how music segments can be stitched together in a plausible way, e.g., Chan et al. (2006). On the other hand, neural networks have been used for music generation since the 80’s (Bharucha & Todd (1989); Mozer (1996); Chen & Miikkulainen (2001); Eck & Schmidhuber (2002)). Mozer (1996) used a Recurrent Neural Network that produced pitch, duration and chord at each time step. Unlike most other neural network approaches, this work encodes music knowledge into the **representation**. Eck & Schmidhuber (2002) was first to use LSTMs to generate both melody and chord. Compared to Mozer (1996), the LSTM captured more global music structure across the song.

早期的方法主要是从音乐理论到生成音乐的知识注入，通过使用音乐片段如何以合理的方式缝合在一起的规则，例如，Chan等人（2006）。另一方面，神经网络已经被用于创作音乐是自80年代起的(Bharucha & Todd (1989); Mozer (1996); Chen & Miikkulainen (2001); Eck & Schmidhuber (2002))。Mozer（1996）用RNN在每一个时间步的持续时间和和弦中产生的音调。与大多数其他神经网络方法不同，这项工作将音乐知识编码为**表示法**。Eck & Schmidhuber (2002)首次使用LSTM创作既有旋律又有和弦的音乐。相比Mozer（1996），LSTM会获取歌曲更加全面的音乐结构。

Like us, Kang et al. (2012) built upon the randomness of melody by trying to accompany it with drums. However, in their model the **scale type** is enforced. No details about the model are presented, and thus it is virtually impossible to compare to. Boulanger-lewandowski et al. (2012) propose to learn complex polyphonic musical structure which has multiple notes playing in parallel through the song. The model is single-track in that it only produces melody, whereas in our work we aim to produce multi-track songs. Just recently, Huang & Wu (2016) proposed a 2-layer LSTM that, like Boulanger-lewandowski et al. (2012), produces music that is more complex than a single note sequence, and is able to produce chords. The main novelty of our work over existing approaches is a hierarchical model that incorporates knowledge from music theory to build the neural architecture, and produces multi-track pop music (melody, chord and drum). We also present two novel fun applications.

像我们一样，Kang等人（的研究建立在）通过尝试用音乐伴随鼓点的方式得来的旋律的随机性。然而，在他们的模型中，**规模类型**被强制执行。没有关于模型的详细信息，因此几乎不可能比较。

Boulanger-lewandowski 等人（2012）建议通过这首歌来学习同时有许多音符被播放的复杂的复调音乐结构。该模式是单一的轨道，因为它只产生旋律，而在我们的工作中，我们的目标是制作多声道歌曲。最近，Huang & Wu (2016)提出了一个两层的LSTM，像Boulanger-lewandowski等人（2012）提出的音乐远比一个单一的音符序列要复杂，音乐是能够产生和弦的。我们对现有方法的研究的主要创新是一个层次模型，它包含了从音乐理论到建立神经结构的知识，并产生了多声道流行音乐（旋律、和弦和鼓）。我们还提出了两个新的有趣的应用程序。

**3．音乐理论概念（CONCEPTS FROM MUSIC THEORY）**

We start by introducing the basic notation and definitions from music theory. A **note** defines the basic unit that music is composed of. Music follows the **12-tone** system, i.e., 12 is the cycle length of all notes. The 12 tones are: E:\study\Intelligent arrangement\12音.jpg A **bar** is a short segment of time that corresponds to a specific number of beats (notes). The boundaries of the bar are indicated by vertical bar lines.

我们从音乐理论的基本符号和定义开始。一个音符定义了音乐构成的基本单位。音乐遵循12音系统，即12是所有音符的周期长度。12个声调是：(如上)。一小节是一个短的时间段，对应于特定数量的节拍（音符）。小节的边界用竖线（小节线）表示。

**Scale** is a subset of notes. There are four types of scales most commonly used: Major (Minor), Harmonic Minor, Melodic Minor and Blues. Each scale type specifies a sequence of relative intervals (or shifts) which act relative to the starting note. For example, the sequence for the scale type Major is 2 → 2 → 1 → 2 → 2 → 2 → 1. Thus, C Major specifies the starting note to be C, and applying the relative sequence of shifts yields:  The subset of notes specified by C Major is thus **C, D, E, F, G, A, and B**[[2]](#footnote-2) (a subset of seven notes). All scales types have a subset of seven notes except for Blues which has six. In total we have 48 unique scales, i.e. 4 scale types and 12 possible starting notes. We treat Major and Minor as one type as for a Major scale there is always a Minor that has exactly the same set of notes. In music theory, this is referred to as Relative Minor.

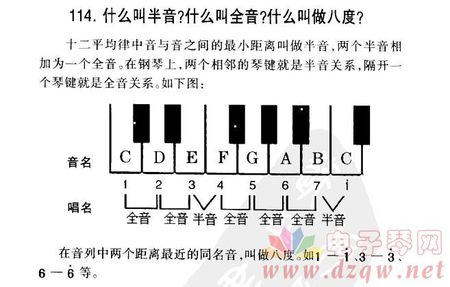
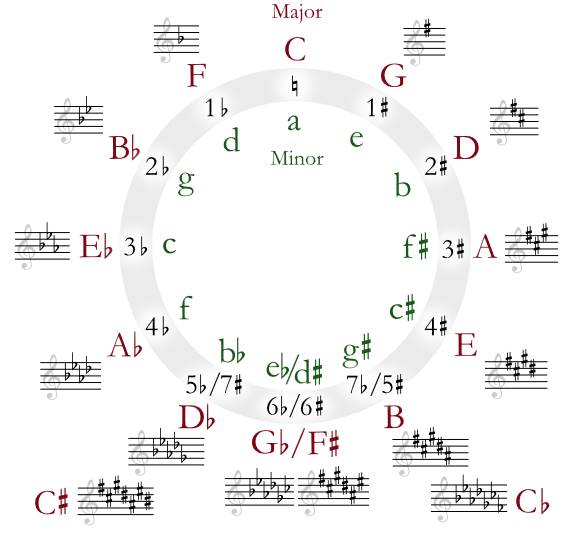
**音阶**是音符的一个子集。音阶最常用的有四种：大调（小调）、和声小调、旋律小调和布鲁斯。每个音阶类型指定与起始音符相对的相对间隔（或移位）序列。例如，大调音阶的序列是:如上。因此，C大调指定起始音为c，并应用移位的相对序列：如上。C大调指定的音符子集是C、D、E、F、G、A和B（七个音符的子集）。所有音阶类型都有7个音符的子集，除了布鲁斯有6个音符。总的来说，我们有48个独特的音阶，即4个音阶类型和12个可能的起始音符。我们把大调和小调作为一种类型，在大调音阶中，总是有一个小调有着完全相同的音符。在音乐理论中，这被称为相对小调。

**Chord** is a group of notes that sound good together. Similarly to scale, a chord has a start note and a type defining a set of intervals. There are mainly 6 types in triads chords: Major Chord, Minor Chord, Augmented Chord, Diminished Chord, Suspended 2nd Chord, and Suspended 4th Chord.

**和弦**[[3]](#footnote-3)是一组听起来很好的音符。类似于音阶，和弦有一个开始音符和一个定义一组音程的类型。在三和弦中主要有6类：大三和弦、小三和弦，增和弦、减和弦、暂停第二和弦、暂停第四和弦。

**The Circle of Fifths** is often used to produce a chord progression. It maps 12 chord starting notes to a circle. When changing from one chord to another chord, moving to a nearby chord on the circle is often preferred as this forms a strong chord progression that produces the sense of harmony.

五度圈通常用来产生和弦的级数。它将12个和弦起始音符映射成一个圆。当从一个和弦转换到另一个和弦时，常常选择在圆周上移动到附近的和弦，因为这形成了一个强大的和弦，产生和谐感。

**** 

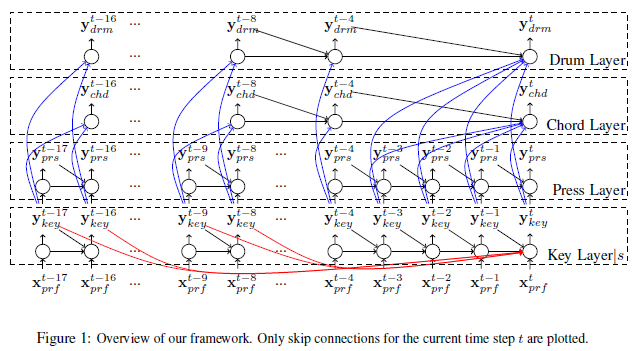
**4流行音乐生成的层次递归网络 （HIERARCHICAL RECURRENT NETWORKS FOR POP MUSIC GENERATION）**

We follow the high level idea behind the Song from π to define our model. In particular, we generate music with a hierarchical Recurrent Neural Network where the layers and the structure of the hierarchy encode our prior knowledge about how pop music is composed. We first outline the model and describe the details and justifications for our choices in the subsections that follow.

我们按照高水平的理念，让这首歌从π定义模型。特别是，我们用分层递归神经网络生成音乐，其中层次和层次结构编码了我们关于流行音乐是如何构成的先验知识。我们首先概述了模型，并描述了我们在后续章节中写了选择的细节和理由。

We condition our generation on the scale type, as this helps the model to pick up the regularities in pop songs. We encode melody with two random variables at each time step, representing which key is being played (the key layer) and the duration that the key will be pressed (the press layer). The melody is generated conditioned on the scale, which does not vary across the song as is typically the case in pop music. We assume the drums and the chords are independent given the melody. Thus conditioned on the melody, at each time step we generate the chord (the chord layer) as well as the drums (the drum layer). The output at all layers yields the final song. We refer the reader to Fig. 1 for an illustration of our hierarchical model.

音阶的类型和我们产生的音乐相适应，这有助于该模型在流行歌曲中找到规律。我们在每一个时间步上用两个随机变量对旋律进行编码，分别表示哪一个键正在播放（键层）和键被按下的持续时间（按压层）。旋律的变化与音阶对应，通常在流行音乐中是一整首歌都不会变的。我们假设鼓和和弦是独立的旋律。因此，在与旋律对应的条件下，在每一个时间步，我们产生和弦（和弦层）以及鼓（鼓层）是一样的。所有层的输出产生最后的歌曲。我们将读者参考图1，以说明我们的层次模型。



图一：框架概述。**只绘制当前时间步长t的跳过连接**

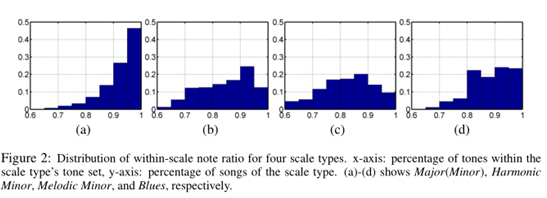
**4.1 音阶的作用（the role of scale）**

It is known from music theory that while in principle each song has 12 tones to choose from, most of the notes are in fact only using the six (for Blues) or seven (for other scales) tone subsets specified by the scale rule. We found that by conditioning the music generator on scale it captures these regularities more easily. However, we do not enforce the notes to be generated from the subset and allow our model to generate notes outside the scale.

从音乐理论上知道，虽然原则上每首歌有12个音调可供选择，但大多数音符实际上只使用六音阶（布鲁斯）或七音阶（其他音阶），音阶所规定的音调子集。我们发现，通过调节音乐发生器的音阶，它更容易捕捉到这些规律。但是，我们不强制从子集生成音符，并允许我们的模型在（我们规定的）音阶之外生成音符。

We confirm the above musical fact by analysing over 100 hours of pop song music from the midi man dataset. Since scale is defined relative to a starting note, we first try to factor out its influence and normalize all songs to have identical start note. To identify the scale of a song, we compute the histogram over the 12 tones and match it with the 48 tone subsets of 4 scale types with 12 different start notes. We then normalize all songs to have start note C by applying a constant shift on all notes. This allows us to categorize any song into 4 scale types. Since this shift affects all notes at once, it does not affect how the song sounds (its harmony). Our analysis shows that for all notes in all Major scale songs, 94.66% are within the tone subset. For Harmonic Minor, Melodic Minor, and Blues the percentage of notes that belong to the main tone set is 87.16%, 85.11%, and 90.93%, respectively. We refer the reader to Fig. 2, where the x-axis denotes the percentage of within-scale notes of a song, and the y-axis indicates how many songs in the dataset have that percentage. Note that the majority of the notes follow the scale rule. Furthermore, different scale types have different inlier distribution. We thus represent scale with a single random variable s ∈ {1,··· ,4} which is fixed for the whole song, and condition the model on **it[[4]](#footnote-4)**.

我们通过分析超过100小时的MIDI音乐数据集中的流行歌曲音乐来证实以上音乐事实。因为音阶是根据起始音符定义的，我们首先要找出影响它的因素，并使所有歌曲正常化，从而获得相同的开始音。为了识别歌曲的音阶，我们计算了12个音调的直方图，并与4个具有12个不同的起始音符的音阶类型的48音子集匹配。然后我们对所有的歌曲进行标准化，通过在所有音符上施加一个常量移位使音符都从音符C开始。这允许我们将任何歌曲分成4种音阶类型。由于这一变化一次性影响所有音符，所以它不影响歌曲的声音（它的和谐）。我们的分析表明，在所有大调音阶中的所有音符中，94.66%个都在音调子集内。对于和声小调、旋律小调和蓝调，属于主基调的音符比例分别为87.16%、85.11%和90.93%。我们将读者引到图2，其中x轴表示歌曲的音阶范围内的百分比，y轴表示数据集中有多少歌曲具有该百分比。注意大多数音符遵循音阶尺度。此外，不同规模类型有不同的层分布。我们因此表示了一个单一的随机变量“s∈{ 1，4 }”，他填充于我们的整首歌并适应模型。

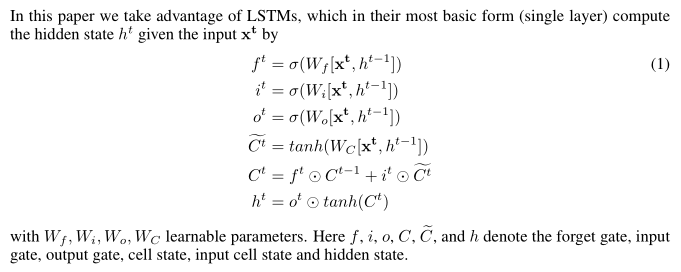


**图二**：四种音阶类型的音阶分布比。x轴：包括音阶类型的音调的百分比；y轴：音阶类型的歌曲的百分比；（a）-（d）分别显示主（小调）、和声小调、旋律小调和布鲁斯。

**4.2 从elody generation中来的两层RNN（TWO - LAYER RNN FORM ELODY GENERATION）**

We represent the melody with two random variables per time step: which key is pressed, and the duration of the press. A Recurrent Neural Network (RNN) is used to generate the key condition on the scale. Then conditioned on the output of the key layer, a second RNN generates the duration of the press at each time step.

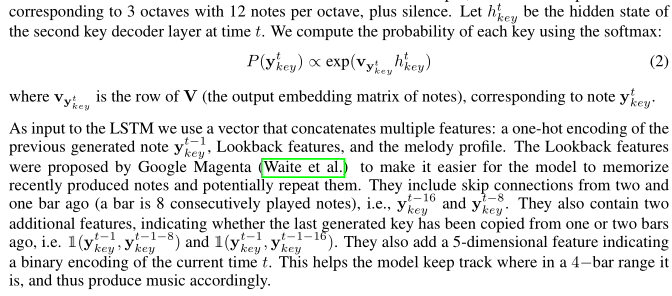
我们在每个步长中用带两个随机变量的旋律来描述：一个是键被按下，另一个是按下的时间长度。一个RNN被用于生成音阶所对应的键。之后对应于按键层的输出，另一个RNN生成在每个时间步长中的按键时间。



在这篇文献中，我们利用LSTMs的优势，用他们最基本的形式（单层）计算“**隐藏态ht**”，通过带有Wf，Wi，Wo，Wc这些可学习参数的（1）式，给定一个输入量“**xt**”的方式。这六个字母分别表示：遗忘门、输入门、输出门、单元状态、输入单元状态和隐藏状态。

In particular, we model the key layer with a two-layer LSTM with 512-dimensional hidden state, which outputs a note (key) at each time step. Note that we condition on scale s, thus we have different parameters per scale. We only allow notes between C3 to C6 as notes outside this range are usually too low or too high to sound good. We remind the reader that given a scale , seven (or six for blues) out of the twelve notes (per octave) are statistically more plausible, however we allow the model to choose from all 12. This results in a 37-dimensional output, as there are 36 possible notes corresponding to 3 octaves with 12 notes per octave, **plus silence**.

特别是，我们模型的**键层（音符？）**有一个512维的隐藏状态的双层LSTM，输出音符（键）是每一个时间步长。注意，我们在音阶s上有条件，所以我们有不同的参数。我们只允许C3到C6之间的音符，因为这个范围之外的音符通常太低或太高，听起来不太好。我们提醒读者，给定音阶，十二个音符中输出七个（或六个蓝色）在统计上是合理的，但是我们允许模型从所有12个音符中选择。这导致在一个37维的输出中，有36种可能的音符对应于3个八度（每个八度有12个音符），**加上沉默**。



把h 当做在时间t处第二个音的解码层的隐藏态。我们用**softmax**计算每个音的可能性：（2），其中Vy是V（音符的输出嵌入矩阵）的行，相当于音符ykey。作为LSTM的输入变量，我们使用一个向量联系多个特点：对先前生成的音符y的一个**热编码，回望特和旋律轮廓。**。回望特征由谷歌Magenta（韦特等人）提出，目的是让我们的模型更容易记住最近生产的音符和潜在的重复。他们包括从两个（音符？）和一小节跳过连接（一小节是8个连续的音符），即xxx和 yyy。他们也包括两个额外的特征，指示最后生成的音是否已从一个或两个之前的小节复制，即（xxx，yyy），（xxx，yyy）。他们还添加了一个五维特征表示一个二进制编码的当前时间，这有助于模型把音轨控制在4小节的范围之内，从而产生相应的音乐。

In addition, we introduce a new feature which we refer to as the **melody profile.** Intuitively, the profile represents the high-level music flow. To get the profile for each song, we compute the local note histogram at each time step with width of two bars, and cluster all local histograms within the song into 10 clusters via k-means. We order the 10 clusters with mean note ordered from low to high as cluster 1 to 10, and apply moving averages on the cluster id sequence to encourage local smoothness. This results in a 10-dimensional one-hot vector representation of the cluster id for each time step. This additional information allows the user to set the melody’s ups and downs of the song.

此外，我们还引入了一个新的特性，我们称之为**旋律轮廓**。直观地说，旋律轮廓代表高级音乐流。为了获得每首歌曲的轮廓，我们在每个时间步中计算出两小节的宽度的本地音符直方图，并通过k-均值将歌曲中的所有局部直方图聚类成10个簇。我们将10个簇的平均值从低到高排列为1到10个簇，并在集群ID序列上移动平均值，以鼓励局部平滑。这将导致每个时间步长的群集ID的10维热矢量表示。这些额外的信息允许用户设定歌曲的旋律起伏。

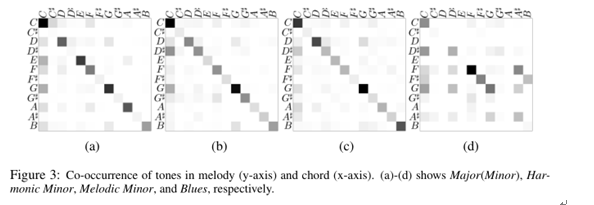
The keys alone are not sufficient to describe how the melody is performed. Additionally we also need to know the duration that each key needs to be pressed for. Towards this goal, conditioned on the melody, we generate the duration of each key with a two-layer LSTM with a 512-dimensional hidden state. We represent the duration of pressing as a forward counting sequence that is conditioned on the generated melody. The press outputs 1 when a new key is pressed, and sequentially outputs 2, 3, 4 and so on as the key is held on. When the current key is released, the press counter is reset to 1. Compared to the event on-off representation of Waite et al., our representation learns the melody flow and how to press separately. This is important, as Waite et al. has extremely unbalanced output distributions dominated by the repeat-of-holding event. We represent press ytprs as a 8-dimensional **one-hot vector**. The input to our LSTM is y t−1prs , concatenated with the 37-dimensional **one-hot encoding** of the melody key y tkey .

单独的键不足以描述旋律是如何执行的。此外，我们还需要知道每个键需要按下的持续时间。为了实现这一目标，使“键”能形成旋律，我们用一个512维隐藏状态的双层LSTM来产生每一个键的时间。我们将持续时间作为一个向前计数序列，以生成的旋律为条件。按下一个新的键时，输出1，并且在键被按住不放时依次输出2, 3, 4等。当当前键被释放时，按下计数器被重置为1。相比于韦特等人“开关”的表现，我们的表现是去学习旋律流如何分别按键。这是很重要的，因为韦特等人具有极不平衡的输出分布，这些分布以重复持续按压某个键的事件为主。我们举例按压ytprs，把它视为一个8维的**热向量。**输入到LSTM是t−1prs，和一个37维旋律键“y tkey”的**热编码**级联在一起。

**4.3 和弦和鼓点的RNN层（CHORD AND DRUM RNN LAYERS）**

We studied all existing chords in our 100 hours of pop music. Although in principle a chord can be any arbitrary combination of multiple notes, we observed that in the actual music data 99.19% of the chords belong to one of 72 chord classes (6 types × 12 start notes). Fig. 3 shows the correlation between the melody’s tone and the starting note of the chord playing at the same time. It can be seen that chord is strongly correlated with melody. These two findings inspire our design. We thus represent chord as a one-hot encoding with 72 classes, and predict it using a two-layer LSTM with a 512-dimensional hidden state. We generate one chord at each time step. The input is concatenated with .

我们在100小时的流行音乐中研究了所有现存的和弦。虽然原则上的和弦可以任意多个音符的组合，但我们观察到，实际数据中和弦音乐的99.19%属于72弦类之一（6种类型×12开始音符）。图3显示了旋律的音调和在同一时间开始弹奏的和弦的起始音符之间的关系。可以看出和弦与旋律有着密切的关系。这两个发现启发了我们的设计。我们因此把和弦作为一个带有72类的热编码，并用一个512维隐藏状态的双层LSTM来预测它。我们在每一个时间步生成一个和弦。输入是，它和与连接在一起。



**图三**：旋律（Y轴）和和弦（X轴）中音调的共同出现。（a）-（d）分别显示主（小调）、和声小调、旋律小调和布鲁斯。

We look at our music dataset and find all unique drum patterns with duration of a half bar. We then compute the histogram of all the patterns. This forms a long tail distribution, where 94.60% comes from the top 100 common patterns. We generate drum conditioned on the key layer using a two-layer LSTM with 512 dimensional hidden states. Drum is represented as one-hot encoding with of the 100 unique **one-bar-long** drum patterns. The input isconcatenated with the notes from the previous three times steps 

我们查看我们的音乐数据集，找到所有独特的有半个小节持续时间的鼓模式。然后我们计算所有模式的直方图。这形成了长尾分布，其中94.60%来自前100种常见模式。我们产生鼓是基于一个512维隐藏状态的双层LSTM的关键层。鼓用一个带有100个独一无二的**单小节长的**鼓点模式的热编码来表示。这次的输入是，它和从前三次步长中得来的音符相连。

**4.4学习（LEARNING）**

We use cross-entropy as our loss function to train each layer. We follow the typical training strategy where we make predictions at each layer and time step but feed in ground-truth information to the next. This effectively decomposes training, and allows to train all layers in parallel. We use the **Adam optimizer**, a learning rate of 2e-3 and a learning rate decay of 0.99 after each epoch for 10 epochs.

我们用交叉熵作为我们的损失函数来训练每一层。我们遵循典型的训练策略，我们在每一层和时间步长上进行预测，但把真实的信息输入下一步。这有效地分解训练，并允许同时训练所有层。我们使用**Adam optimizer**，一个2e-3的学习速率和一个经历了10个时期后衰减0.99的学习速率

**4.5音乐合成：把所有的输出放在一起（MUSIC SYNTHESIS : PUTTING ALL THE OUTPUTS TOGETHER）**

To synthesize music we first randomly choose a scale and a profile. For generating, we randomly choose one cluster id with a random duration, and repeat until we get the desired total length of the music sequence. We then perform inference in our model conditioned on the chosen scale, and use as input to our key layer. At each time step, we sample a key according to. We encode it as a one-hot vector and pass to the press, chord and drum layers. We sample the **press**, chords and drums at each time step in a similar fashion.

为了合成音乐，我们首先随机选择一个比例和一个侧面。为了生成，我们随机选择一个随机时间的群集ID，并重复，直到我们得到音乐序列的所需的总长度。然后，我们在我们所选择的尺度的条件下执行推理，并使用作我们的关键层的输入。在每个时间步骤中，我们根据来抽样一个键。我们将其编码为一个热矢量，并传递给**press**、弦和鼓层。我们在每一个时间步上按同样的方式采样**press**、和弦和鼓声。

Before putting the outputs across layers together, we further adjust the generated sequences at the bar level. For melody, we first check at each bar if the first step is a continuation of a previous note or silence. If it is the latter, we find the first newly pressed note within the bar and move it to the beginning of the bar. We do similarly for the windows of two half-bars as well as the four quarter-bars. This makes the melody more likely to be on the beat, and generally sounds better. We verify this in our experiments.

在将输出跨层的放在一起之前，我们在小节这个等级上进一步调整生成的序列。对于旋律，我们首先检查每一个小节，是否第一步是一个前一个音符或休止的延续。如果是后者，我们会在这个小节里找到第一张新的奏出的音符，并将它移到该小节的开头。我们对两个二分之一小节和四个四分之一小节做同样的事。这使得旋律更可能在节拍上，通常听起来更好。我们在实验中验证了这一点。

For chord, we generate one chord at each half bar, which is the majority of all single step chord generations. Furthermore, we incorporate the rule of chord progression in the *Circle of Fifths* as between chords pairwise smooth terms, and compute the final chord using dynamic programming. For drum, we generate one pattern at each half bar.

对于和弦，我们在每半个小节上产生一个和弦，这是大部分的所有单阶弦的生成。此外，我们把*Circle of Fifths*中的弦程规则作为和弦成对平滑项之间的规则，并用动态规划计算最后的和弦。对于鼓，我们在每半个小节上产生一个模式。

Our model generates with scale starting note C, and then applies a constant shift to generate music with other starting notes. Besides scale, which instrument to use is also customizable. However, we simply set all instruments as grand piano in all experiments, as the effect and musical meaning of different instrument combinations is beyond the scope of this paper.

我们的模型音阶的生成开始于音符C，然后应用一个恒定的转变，以产生音乐与其他起始音符。除了音阶外，使用哪种仪器也是可定制的。然而，在所有实验中，我们仅仅将所有乐器都设置为大钢琴，因为不同乐器组合的效果和音乐意义超出了本文的范围。

**5.实验（E XPERIMENTS）**

To train our model, we took 100 hours of pop music from midi  **man** which consists of user-composed pop songs and video game music. In our generation, we always use 120 beats per minute with 4 time steps per beat. However, songs in the dataset can have arbitrary speed. To neutralize the effect of this, we detect the most frequent interval between two adjacent notes for each song, and iteratively divide or multiply this interval by 2 until it falls in the range between 0.25s and 0.5s. We use this as a measure of the song’s beat duration. We then adjust the song’s temporal axis so that all songs have the same beat duration of 0.5s.

为了训练我们的模型，我们花了从MIDI man中得到的 100个小时的流行音乐，这些音乐由用户组成的流行歌曲和视频游戏音乐组成。在我们生成的过程中，我们总是使用每分钟120拍，每拍4个时间步长的。但是，数据集中的歌曲可以有任意的速度。为了消除这种影响，我们检测到每首歌曲最常见的两相邻音符之间的间隔，并反复除以或乘以这个区间2，直到它落在0.25s和0.5S级之间的范围。我们用这个作为衡量歌曲的节拍时间。我们用这个作为衡量歌曲的节拍时间。然后调整歌曲的时间轴，以便于所有的歌都有相同的节拍，时间为0.5s。

A MIDI file can be separated into different channels/tracks, where the 9th channel is specifically preserved for drums. We categorize the rest of non-drum tracks into melody, chord, and else, by simply setting thresholds on average number of unique notes within a bar and average number of note changing within a bar, as chords are by definition repetitive. Fig. 4 shows an example of our music generation.

MIDI文件可以分成不同的通道/磁道，其中第九个通道专门为鼓保存。我们将剩下的非鼓形曲目分类为旋律、和弦等，只需设置一个小节内独特音符的平均数和这个小节内平均音符数的变化即可，因为和弦会按定义重复。图4显示了我们音乐生成的一个例子。

To evaluate the quality of our music generation, we conduct a human survey with 27 participants. In the survey, participants are presented with several pairs of 30-second music clips, and are asked to vote which clip in the pair sounds better. We gave no other information about what they are listening to. They are also allow to submit a neutral vote in case they cannot decide between the two choices. In our study, we consider three cases: our full method versus Magenta Waite et al., our **method** with melody only versus Google Magenta (Waite et al.), and our method versus our method without the temporal alignment described in Sec.4.5. We randomly generated 10 songs per method and randomly shuffled each pair.

为了评估我们音乐创作的质量，我们进行了一项27人参加的人类调查。在这项调查中，参与者们会得到几对30秒的音乐片段，并被要求投票选出每对音乐中哪个片段更好。关于他们在听什么，我们没有提供任何其他信息。他们也允许提交中立的投票，以防他们不能在两种选择之间作出决定。在我们的研究中，我们考虑三例（三种比较）：我们的完整**方法**与Magenta Waite等人的方法；我们只有旋律的方法和旋律与Google Magenta （Waite等人）；我们的方法和没有经过4.5节描述中是时间对准的我们的方法。我们每个方法随机生成10首歌曲，并随机调整每对歌曲。

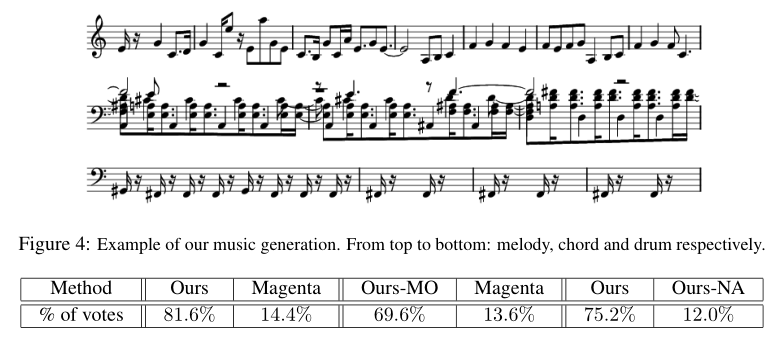


Table 1: Human evaluation of music generated by different methods: ours and Waite et al.’s Magenta. Ours-MO and Ours-NA are short for Ours Melody Only and Ours No Alignment. We allowed neutral votes, thus the sum of the pair is less than 100%.

表1：用不同的方法产生的音乐的人耳评价：我们和Waite等人的Magenta。Ours-MO和Ours-NA分别是Ours Melody Only和Ours No Alignment的简称。我们允许中立票数，因此这对的总和小于100%。

As shown in Table 1, most participants prefer songs produced by our method compared to Magenta. Participants also made comments such as *music sounds better with percussion than piano alone*, and *multiple instruments with continuous play is much better*. This confirms that our multi-layer generation improves music quality. Few participants also point out that *drums sound too different and do not participate to the melody perfectly*, which indicates that further improvements can be still made. In the second comparison, we study if the quality improvement of our method is only caused by adding chords and drums, or is also related to our two-layer melody generation with alignment. It can be seen that without chords and drums, the score drops as expected, but is still much higher than the Magenta baseline. This is because our method *produces less recursion and silence*, and *faster and more accurate tempo* as mentioned by the participants. In the last comparison, most participants prefer our full method than the no-alignment version, since *beats are more subtle and better timed*. This proves the usefulness of temporal alignment.

如表1所示，大多数参与者喜欢用我们的方法制作的歌曲，而不是Magenta。参与者还发表了意见，带有打击乐的声音要比单独的钢琴声更好，多个乐器连续播放就更好了。这证实了我们的多层生成提高了音乐质量。也很少有人指出鼓的声音太大，并不能完美地参与到旋律中来，这意味着仍然可以作出进一步的改进。在第二个比较中，我们研究了我们的方法的质量改进是否只由增加和弦和鼓引起，或者还是与我们的两层旋律生成对齐有关。可以看出，如果没有和弦和鼓点，乐谱会像预期的那样下降，但仍比Magenta基线高出很多。这是因为我们的方法产生较少的递归和休止，并且产生了更多精确的节奏，就像参与者所提到那样。在最后一个比较中，大多数参与者更喜欢我们的完整方法，而不是没有对齐的版本，因为节拍更精细，时间更好。这证明了时间对齐的有用性。

Finally we study our model’s capabilities to generate new music. Towards this goal, we generated

100 sequences of 50 seconds of length using different random initializations. Then for each sequence, we search for the longest sub-sequence of keys that matches part of the training data, and record its length. We find out that with 1 hour of training data, the mean matching sub-sequence length is 3.46s. With 100 hours of training data, the mean length increases to 4.65s, since there are more possible matches. The sequences are very small and thus, our model is able to generate new music.

最后，我们研究我们的模型生成新音乐的能力。为了实现这一目标，我们生成长度为50秒的100个序列，使用不同的随机的初始化。然后，对于每个序列，我们搜索匹配于训练库中的部分数据中的最长子序列，并记录这些序列的长度。我们发现，随着训练数据1小时，平均匹配的子序列长度为3.46s。训练100小时，平均长度增加到4.65s，因为有更多的可能的匹配。序列非常小，因此，我们的模型能够产生新的音乐。

**6.应用（6APPLICATIONS）**

In this section we demonstrate two novel applications of our pop music generation framework. We refer the reader to http://www.cs.toronto.edu/songfrompi/ for the music videos.

在本节中，我们将为读者演示了流行音乐生成框架的两个新应用。我们推荐读者去<http://www.cs.toronto.edu/songfrompi/>下载我们的音乐视频。

**6.1 NEURAL DANCING AND KARAOKE**

In our first application, we attempt to generate both music and a stickman dancing to it, as well as a sequence of karaoke-like text that people can sing along with. To learn the relationship between music and dance, we download 1 hour of video from the game Just Dance, as well as the MIDI files for songs included in the video from different sources. We use the method in Newell et al. (2016) to track single-frame 2D human pose in the videos. We process the single-frame tracking result to ensure left-right body consistency through time, and then use the method of Zhou et al. (2016) to convert the 2D pose sequence into 3D. Example results are shown in **Fig. 5**. We observe that our pose processing pipeline is able to extract reasonable human poses most of the time. However, the quality is not perfect due to tracking failure or video effects. We define pose similarity as average euclidean distance of all joints, and cluster poses into 456 clusters. **We used Frey & Dueck (2007) as the number of clusters is large.**

在我们的第一个应用中，我们试图同时产生音乐和火柴人跳舞给它，以及一系列类似卡拉文本，就是人可以一起跟着唱的那种。为了学习音乐和舞蹈之间的关系，我们从游戏“Just Dance”中下载了1小时的视频，以及包含来自不同来源的视频的歌曲的MIDI文件。我们使用Newell等人的方法（2016）在视频中跟踪单帧2d人体姿势。我们对单帧跟踪结果进行处理，以保证左右体的一致性，然后使用Zhou等人的方法（2016）将2D位姿序列转换为3d。例结果如图5所示。我们观察到，我们的姿势处理管道能够在大多数时间内提取出合理的人体姿势。然而，由于跟踪失败或视频效果，质量并不理想。我们把所有关节的平均欧氏距离定义为姿势相似性，并把集成出的姿势分成和456个集群。我们用Frey和Dueck（2007）作为集群的数量。

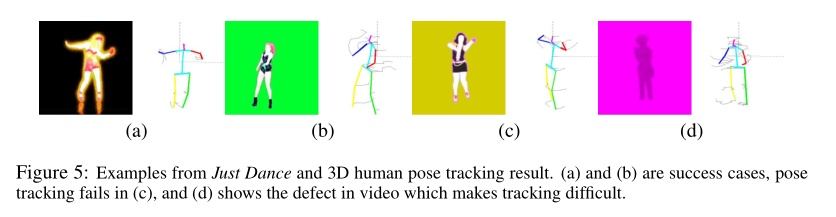


图5：来自‘Just Dance’和3D人体姿势轨迹结果的事例。（a）和（b）是成功的事例。（c）中的姿势轨迹发生错误，（d）表示出视频的缺陷造成轨迹的识别困难。

We learn to generate a stickman dancing by adding another dancing layer on top of the key layer, just like for drum and chord. We generate one pose at each beat, which is equivalent to 4 time steps or 0.5 seconds in a 120 beat-per-minute music. In particular, we predict one of the 456 pose clusters using a **linear projection** layer followed by softmax. We use cross-entropy at each time step as our loss function. At inference time, we further apply moving average to temporally smooth the generated 3D pose sequence.

我们学会了通过在键层顶部添加另一哥舞蹈层来让一个火柴人跳舞，（方法）就像对鼓与弦一样。我们在每拍产生一个姿势，这相当于在每分钟120拍的音乐中的4个时间步长或0.5秒。特别是，我们预测456个姿势集群之一是使用一个**线性投影层**通过SOFTMAX。我们在每个时间步长使用交叉熵作为损失函数。在推理时间，我们进一步应用移动平均值去暂时地平滑生成的三维姿态序列。

To learn the relationship between music and lyrics, we collect 51 hours of lyrics data from the internet. This data contains 50 hours of text without music, and the rest 1 hour are songs we collected from Just Dance. For the music part, we temporally align each sentence in the lyrics with the midi music by using the widely-existing lrc format, which records the time tag at the beginning of every sentence. We select words that appear at least 4 times, which yields a vocabulary size of 3390 including unknown and end-of-sentence. Just as for dance, we generate one word per beat using another lyrics layer on top of the key layer.

为了学习音乐和歌词之间的关系，我们从互联网上收集51小时的歌词数据。这个数据包含50小时的音乐文本，剩下的1小时是我们从舞蹈中收集的歌曲。对于音乐的一部分，我们暂时利用广泛存在的LRC格式，来排列MIDI音乐中每个句子的歌词，（LRC格式会在每个句子的开头记录时间的标签）。我们选择至少出现4次的词，产生3390个词汇量，它包括未知的和句子结束。就像舞蹈一样，我们在键层的顶部使用另一个歌词层来生成每拍一个单词。

**6.2 NEURAL STORY SINGING（神经故事歌唱）**

In this application our aim is to sing a song about a photo. We first generate a story about the photo with the neural storyteller Kiros et al. (2015) and try to accompany the generated text with music. We utilize the same 1 hour dataset of temporally aligned lyrics and music. We further include the phoneme list of our 3390 vocabulary as we also want to sing the story. Starting from the text produced by neural storyteller, we arrange it into a temporal sequence with 1 beat per word and a short pause for end-of-sentence, where the pause length is decided such that the next sentence starts from a new bar. As our dataset is relatively small, we generate the profile conditioned on the text, which has less dimensions compared to the key. This is done by a 2-layer LSTM that takes as input the generated profile at the last time step concatenated with a one-hot vector of the current word, and outputs the current profile. We then generate the song with our model given the generated profile. The generated melody key is then used to decide on the pitch frequency of a virtual singer, assuming the key-to-pitch correspondence of a grand piano. We further constrain that the singer’s final pitch is always in the range of E3 to G4, which we empirically found to be the natural pitch range. We then replace all words outside the vocabulary with the sound Ooh, and play the rendered singing with the generated music.

在这个应用程序中，我们的目标是唱一首关于照片的歌曲。我们首先生成一个与神经的说书人奇洛斯等人的照片故事（2015）试着用音乐伴奏生成的文本。们利用相同的1小时数据集的时间对齐歌词和音乐。我们还包括我们3390个词汇的音素列表，因为我们也想唱这个故事。从神经说书人产生的文本开始，我们把它安排成一个时间序列，每个单词有1个节拍，一个短暂的停顿以结束句子，停顿的长度取决于了下一个句子开始的新的小节。由于我们的数据集相对较小，所以我们生成了与文本相关的概要文件，与关键字相比，它的维度更少。这是由两层LSTM作为输入生成的轮廓在最后时间步级联一个火热的矢量字当前，输出电流波形。然后，我们根据生成的概要文件用我们的模型生成歌曲。然后用生成的旋律键决定一个虚拟歌手的音高频率，假设音调是一架大钢琴的音调。我们进一步限制，歌手的最后一节总是在E3 G4的范围，我们的经验是自然的音高范围。然后，我们用声音的声音替换词汇表以外的所有单词，并用生成的音乐播放渲染的歌曲。

**7 结论和未来工作（CONCLUSION AND FUTURE WORK）**

We have presented a hierarchical approach to pop song generation which exploits music theory in the model design. In contrast to past work, our approach is able to generate multi-track music. Our human studies shows the strength of our framework compared to an existing strong baseline. We additionally proposed two new applications: neural dancing & karaoke, and neural story singing.We next discuss the limitations and avenues for future work. As most existing approaches our method’s objective is to learn to produce music at the note level. This can be unsuitable for music, as music is flexible and intentionally made to be unpredictable when it is composed. This calls for a deeper study of music theory, as in this paper we are only scratching the surface.

在模型设计中，我们提出了一种基于音乐理论的流行歌曲生成方法。与过去的工作相比，我们的方法能够产生多音轨的音乐。与现有的强基线相比，我们的人类研究显示了我们框架的优势。我们还提出了两个新的应用：①神经舞蹈和卡拉OK，②神经故事唱歌，我们接下来讨论局限性和未来工作的途径。正如大多数现有的方法一样，我们的方法的目标是学会在音符级别上产生音乐。这可能不适合音乐，因为音乐是灵活的，故意使它在组成时是不可预测的。这就要求对音乐理论进行更深入的研究，因为在这篇文章中，我们只是在触及表面。

**参考文献REFERENCES**

1. Jamshed J. Bharucha and Peter M. Todd. Modeling the perception of tonal structure with neural nets. Computer Music Journal, 13(4):44–53, 1989.
2. Nicolas Boulanger-lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription.In ICML, 2012.
3. Michael Chan, John Potter, and Emery Schubert. Improving algorithmic music composition with machine learning. In 9th International Conference on Music Perception and Cognition, 2006.
4. Chun-Chi J. Chen and Risto Miikkulainen. Creating melodies with evolving recurrent neural networks. In International Joint Conference on Neural Networks, 2001.
5. Douglas Eck and Juergen Schmidhuber. A first look at music composition using lstm recurrent neural networks. 2002.
6. Steve Engels, Fabian Chan, and Tiffany Tong. Automatic real-time music generation for games. In AIIDE Workshop, 2015.【没找到】
7. Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. Volume 315, pp. 972–976, 2007.
8. Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. In arXiv:1508.06576, 2015.
9. Allen Huang and Raymond Wu. Deep learning for music. arXiv preprint arXiv:1606.04930, 2016.
10. Daniel Johnson. Composing music with recurrent neural networks. <https://goo.gl/YP9QyR>.

【http://www.hexahedria.com/2015/08/03/composing-music-with-recurrent-neural-networks/】

1. Semin Kang, Soo-Yol Ok, and Young-Min Kang. Automatic Music Generation and Machine Learning Based Evaluation, pp. 436–443. Springer Berlin Heidelberg, 2012.【没找到】
2. Andrej Karpathy, Justin Johnson, and Li Fei-Fei. Visualizing and understanding recurrent networks.In ICLR 2016 Workshop, 2016.
3. Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In NIPS, 2015.
4. David Macdonald. Song from \_. <https://youtu.be/OMq9he-5HUU>.**【网页里】**
5. Reddit midi man. Midi collection. <https://goo.gl/4moEZ3>.**【网页里】**
6. Michael C. Mozer. Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing. Connection Science, 6(2-3), 1996.
7. Alejandro Newell, Kaiyu Yang, and Jia Deng. Stacked hourglass networks for human pose estimation. In ECCV, 2016.
8. Arnold Schoenberg and Dika Newlin. Style and idea. Technical report, Williams and Norgate London, 1951.【没找到】
9. Edgar Simo-Serra, Sanja Fidler, Francesc Moreno-Noguer, and Raquel Urtasun. Neuroaesthetics in fashion: Modeling the perception of beauty. In CVPR, 2015.
10. Felix Sun. Deephear - composing and harmonizing music with neural networks. <https://goo.gl/7OTZZL>.**【网页里】**
11. Elliot Waite, Douglas Eck, Adam Roberts, and Dan Abolafia. Project magenta. <https://magenta.tensorflow.org/>. **【网页里】**
12. Xiaowei Zhou, Menglong Zhu, Spyridon Leonardos, Kosta Derpanis, and Kostas Daniilidis. Sparseness meets deepness: 3d human pose estimation from monocular video. In CVPR, 2016.

1. https://youtu.be/OMq9he-5HUU [↑](#footnote-ref-1)
2. c#就是在c的基础上向上升高小二度。将音符升高半音，叫升音。用“#”（升号）表示。标准的音降低半音，用” b "(降号）表示。基本音升高全音叫重升音，用“x"(重升）表示。基本音降低全音叫重降音。用“ bb”（重降）表示。 [↑](#footnote-ref-2)
3. 三个或三个以上，按照一定度数关系排列起来的一组音，称为和弦。和弦中的各音之间是一般三度关系，但也有不按三度关系的。不同的和弦具有不同的色彩属性，可以达到不同的声音效果，这使得和弦的配置成为音乐理论中十分重要的一项内容。和弦一般是三和弦，其从低到高三个音分别称为根音、三音、五音。如果是七和弦的话，就多了个七音。 [↑](#footnote-ref-3)
4. 对于有音乐背景的读者，十二音序列技法，Schoenberg & Newlin (1951) 防止单一强调任何一个音阶。然而，我们的数据分析表明流行音乐不受它的影响。 [↑](#footnote-ref-4)