A Comparative Study of LSTM and Transformer Models in Music Melody Generation

Jiaxiang ZHENG[1] Moxi CAO[2]

[1, 2]Kangwon National University, Chuncheon, 243411, Korea

ABSTRACT

[Background]In recent years, using deep learning models to generate music has become the mainstream direction in AI music. However, the main models for music generation still face several issues, the biggest of which is the inability to effectively simulate musical structure, hindering computers from creating compositions that conform to musical structures. **[Objective]** To address this, we need to explore which models can effectively simulate the structure of music and create more humanized music. **[Method]**We conduct comparative experiments, analyzing the advantages and disadvantages of music generated by LSTM and Transformer models, and propose improvements based on the findings. **[Results]** Experimental results demonstrate that LSTM performs better than Transformer in simulating musical structure in shorter sequences, but struggles with longer sequences; whereas Transformer outperforms LSTM in handling longer sequences and can effectively simulate musical structure in longer sequences after improvements, creating compositions that align with human musical perception. **[Conclusion]**Therefore, we believe that the Transformer model is more suitable for AI music composition tasks, and improving its attention mechanism to enhance recognition of musical structure will be the mainstream direction for music generation in the future.

Keywords
AI Composition; Music Generation; Deep Learning; Transformer

Corresponding author
Moxi CAO
ORCID: 0009-0000-2769-2316
1965387584@qq.com

Received: 01. Nov. 2023
Reviewed: 20. Dec. 2023
Accepted: 25. Dec. 2023

DOI: 10.23112/jgas23123102

Editor:Jiayong YU

LSTM 与 Transformer 模型在音乐旋律生成任务上的比较

郑嘉祥 [1] 郑嘉祥 [2]

[1,2] 江原大学，春川， 24307，韩国

摘要

【背景】近年来，利用深度学习模型生成音乐已经发展为 AI 音乐的主流方向，但音乐生成任务的主流模型仍然存在着一些问题，其中最大的问题就是不能有效地模拟音乐结构，使计算机创作出符合音乐结构的乐曲。**【目的】**为此，我们需要探究哪些模型能够很好地模拟音乐的结构，创造更加人性化的音乐。**【方法】**我们通过对比实验，对比 LSTM 与 Transformer 模型所生成音乐的优点与缺点，并在此基础上提出改进方案。**【结果】**实验结果证明，LSTM 在较短的序列上模拟音乐结构的表现优于 Transformer，但其无法处理过长的序列；而 Transformer 在处理较长序列的表现优于 LSTM，并通过改进后能在较长的序列上有效地模拟音乐结构，创作出符合人类音乐听觉的乐曲。**【结论】**因此我们认为，Transformer 模型更加适合 AI 音乐作曲任务，并在未来通过改进其注意力机制来提高音乐结构的识别能力是音乐生成的主流方向。

Keywords

AI 作曲 ; 音乐生成；深度学习；Transformer

Corresponding author

曹墨曦

ORCID: 0009-0000-2769-2316

1965387584@qq.com

Received: 01. Nov. 2023

Reviewed: 20. Dec. 2023

Accepted: 25. Dec. 2023

DOI: 10.23112/jgas23123102

Editor:Jiayong YU

1 前言

AI 作曲是以计算机通过特定的程序自创作音乐的新兴技术。随着人工智能的深度学习模型在计算机视觉（CV）、自然语言处理（NLP）领域的成功，人们看到了其在其他领域上的可能性，其中包括利用 AI 作曲，又称音乐生成。文本与符号音乐有着天然的相似性，两者在时序性的特点上表现出高度同一的形式，因此学界看到了以 NLP 领域的模型来生成音乐的未来。

2.研究背景

2.1 符号音乐

符号音乐是指使用音乐符号来表示音乐的一种方式，当下最流行的五线谱就是符号音乐的形式之一，其源头可追溯至 11 世纪。音乐符号包括音符、节拍、和弦、音高、音长等元素，它们以特定的符号和符号排列方式来表示音乐的内容和结构。符号音乐是传统音乐领域中的主要表达方式，它使音乐可以被演奏、传承和记录。符号音乐是计算机音乐出现前最主流的音乐记录方式。

随着科技的发展，音乐开始以数据的形式在计算机中存储，其中包括波形形式与 MIDI 为代表的符号形式。MIDI 能够实现传统乐谱记录音乐音高、时值、速度的功能，还能够表达传统乐谱不能记录的信息，例如音符力度、音乐表情、声向等。此外，其占用的内存空间非常小，通常是波形表示的百分之一以下。因此，MIDI 格式被广泛运用于音乐生成领域中。

2.1 符号音乐

音乐生成任务是指使用人工智能技术来创造音乐的过程。这种任务可以包括从头开始生成全新的音乐作品，也可以是根据已有的音乐或音乐片段进行改编和生成新的作品。音乐生成任务通常涉及以下一些方面：

（1）风格化生成：这是生成具有特定音乐风格或风格特征的音乐的任务，模型可以被训练以模仿不同的音乐流派，如古典、流行、爵士、摇滚等。

（2）自动作曲：自动作曲是一种生成原创音乐的任务，它可以根据一些输入条件，如音乐理论规则、和弦进程或旋律模式，自动生成新的音乐作品。

（3）音乐伴奏生成：这种任务可以为旋律生成多乐器的伴奏音乐。

（4）音乐改编：它可以采用已有的音乐片段，并对其进行改编或变换，以生成新的音乐作品。这可以包括将一首古典音乐作品转化为流行风格，或者将一首歌曲的节奏加速等。

（5）音乐生成技术还可以应用于电子游戏、电影制作、广告音乐、背景音乐、教育以及音乐创作的辅助工具中。

音乐生成任务通常涉及使用机器学习、深度学习和自然语言处理等人工智能技术。这些技术可以帮助计算机程序学习音乐的结构、模式和风格，并生成与人类创作的音乐相似的作品。这一领域的研究和发展正在不断扩展，使得计算机生成的音乐变得越来越接近人类创作的音乐，为音乐产业和音乐创作带来了新的可能性。

2.3 相关工作

音乐生成领域的发展是以深度学习领域的

进步为基础的，从早期的 CNN、RNN 到今天的 Transformer[2-5][14-17]，音乐生成领域可借用的工具越来越多。虽然音乐生成的方法一直在进步，但其创作的音乐却始终没有达到人类创作音乐的水准。但即使如此，前人进行的探索与提出的方法也是非常宝贵的经验，其中主要包括：

2.3.1 传统 RNN

在 (Zhu et al.,2018) 中提出了一种流行音乐的旋律和编曲生成框架。该框架采用了循环神经网络（RNN），具体来说是门控循环单元（GRUs）(Zhu, H., Chen, E., Liu, Q., Yuan, N. 等，2018），用于生成带有和弦进行的旋律。与此同时，研究人员设计了一个多乐器协同编曲模型，这个模型利用了更新的 GRUs，以支持多任务学习，从而实现多轨音乐编曲。

2.3.2 LSTM

LSTM 是一种特殊类型的 RNN，其目的是解决传统 RNN 的长期依赖问题。它被设计用于避免反向传播误差的快速衰减。DeepJ，是一种基于 LSTM 的音乐生成模型，用于生成具有特定风格的音乐。DeepJ 的训练过程使用了钢琴卷音符表示，并采用了双轴长短时记忆（LSTM）(Zhu, H., Chen, E., Liu, Q., Yuan, N. 等，2018），以适应三个主要古典音乐时期（巴洛克、古典和浪漫）的音乐风格。DeepJ 能够在混合条件下生成符合特定作曲家风格的音乐。

2.3.3 自注意机制与 Transformer

相较于传统的 RNN，基于自注意机制（多头）的序列模型 Transformer（Vaswani, A., Shazeer, N., Parmar. 等 ,2017）在训练和推断方面具有更好的并行性和可解释性。Transformer 在需要保持长距离连贯性的任务中取得了卓越的结果，包括神经机器翻译、预训练语言模型、文本到语音合成以及语音识别等领域。但其注意力机制所导致的二次复杂度依旧是一个挑战。

2.3.4 Music Transformer

Music Transformer（Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Simon, I. 等 ,2018) 是一种基于 Transformer 的模型，采用相对注意机制，用于生成具有长期结构的钢琴音乐。它采用绝对时间间隔来表示音符序列，以确保音乐具有长距离的相关性。这个模型的独特之处在于，它可以从作曲家书写的乐谱中学习音符的组合方式，这对于音乐创作非常重要。但其问题是无法生成符合真实演奏的音乐。

2.3.5 最新的音乐生成 Transformer

符号音乐生成的学者们提出一些新的对于 MIDI 标记的表示方法，旨在以较少的音乐标记来表示音乐，以降低音乐序列的长度，其中包括复合词表示法（Hsiao, W.-Y., Liu, J.-Y., Yeh, Y.-C. 等，2021) 与 OctupleMIDI (Zeng, M., Tan, X., Wang, R. 等 ,2021) 等。这种方法有效地减少了音乐序列表示的长度，但是对于 Transformer 来说，完整的音乐序列还是太长，这也会大大地限制生成音乐的长度。

此外，还有一条路径是使用长序列的改进版 Transformer 作为基础模型成音乐，其中包括 Transformer-XL（Dai, Z., Yang, Z., Yang, Y. 等，2019）、Linear-Transformer（Katharopoulos, A., Vyas, A., Pappas, N. 等 ,2020)。尽管这些模型可以处理更长的序列，但由于这些模型最初用于自然语言处理领域，因此无法抽象于模拟音乐的组织结构（例

如重复、模进等）。

这个问题在 2021 年微软亚洲研究院所提出的 Museformer（Yu, B., Lu, P., Wang, R. 等，2022) 模型得以解决。Museformer 通过改进注意力模块，提出“粗粒度”与“细粒度”的注意力机制，使其可以有效地抽象乐句级曲式结构上的重复，并且能够根据音乐的发展合理地安排音乐的乐段。尽管如此，但其对于模拟音乐的发展手法（如乐汇级的重复与模进）表现并不是很出色，这也是近年来在该领域有待解决的主要问题之一。

2.4 本文目标

本文旨在通过 LSTM 与 Transformer 模型的对比研究，分析两者所存在的问题，并尝试通过改进这些问题来生成更加人性化、更加符合音乐规则的乐曲。

3.研究方法

3.1 相关音乐理论

音乐是时间的艺术。关于音乐的要素，学界有着几种不同的观点。首先，从声学角度来看，音乐可以通过其时间、波形和频率特征来描述。然而，从音乐角度来看，音乐的基本要素包括音高、音长、音量、音色、和空间位置等。不过，音乐中最重要的要素是音高和音长，两者通过特定的组织结构构成旋律。

音乐的旋律存在着一定的组织结构与周期性，它是某个特定的动机通过一系列的音乐发展手段组织为乐节、乐句、乐段，直至整首乐曲。虽然，音乐组织结构的最小单位为单个音符，但是实际上，其在音乐中是以乐汇为最小结构单位的。

如果我们将音乐旋律类比于文字，那么单个音符就是“字”，乐汇就是“词”，而乐句就是一个完整的句子。因此，借助 NLP 领域成熟的深度学习模型来模拟音乐“词”与“句”的组织关系就是符号音乐生成的关键。

3.2 数据表示

为了使音乐能够被深度学习模型处理，我们需要将音乐表示为深度学习可以理解的语言。首先，我们需要将数据集内的乐曲都移调至 C 大调（或 a 小调），这样做的好处是所有乐曲都能够表示成音级的形式。然后，我们将数据集内的整首乐曲划分为 n 个乐汇向量 x，再将 n 个乐汇装进一个矩阵 X，由此可得一首乐曲的表示为 $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \dots, \boldsymbol{x_n}]^T$。

接下来，我们定义单个乐汇 x 内的音符的表示方式。我们需要将处理好的 MIDI 文件表示为 1 ~ 52 中的整数，用来代表钢琴上所有白键（C 大调的自然音级）的音高，于是得到单个乐汇的表示方式 $\boldsymbol{x} = [a_1, a_2, \dots, a_m]$，其中 a 为单个音符的音高，m 为 x 包含音符的数量。这样处理的原因是，所有相邻音级之间的音程距离就可以在数字上表示为等距的形式，而音乐的组织结构正是依靠这种形式才得以构建的。

除音高外，还需要将节奏表示成相应的值嵌入 x 内。我们以 1 表示音乐中的四分音符，则可以得到 0.5 为八分音符、0.25 为十六分音符、2 为二分音符、4 为全音符。我们将乐汇中所有音符的持续时间用向量 r 表示，则 $\boldsymbol{r} = [r_1, r_2, \dots, r_m]$ 。最后将 r 嵌入 x 中，得到包含节奏与音高值的乐汇 x。

3.3 相似度判定

音乐的发展依赖于基于重复规则的模进、倒影等发展手法，因此本节则以数学的形式来实现模进、重复的数学表示。

令乐汇$\boldsymbol{a}=[\boldsymbol{a_1},\boldsymbol{a_2},\ldots,\boldsymbol{a_n}]$，乐汇$\boldsymbol{b}=[\boldsymbol{b_1},\boldsymbol{b_2},\ldots,\boldsymbol{b_n}]$，若 a 与 b 呈模进关系，则式（1）成立。

$$\boldsymbol{a_1}-\boldsymbol{b_1}=\boldsymbol{a_2}-\boldsymbol{b_2}=\cdots=\boldsymbol{a_n}-\boldsymbol{b_n}$$

从式(1)可以看出，如果a与b呈模进关系，则 $\boldsymbol{a}-\boldsymbol{b}$ 中的每一个元素都相等。如果 $\boldsymbol{a}-\boldsymbol{b}$中每一个元素均为 0，那么 a 与 b 呈重复关系。

但我们同时发现，式（1）仅适用于严格模进的情况，于是对于变化模进，我们定义了式（2）：

$$\boldsymbol{M(a,b)}=\frac{\boldsymbol{S(a-b)}}{\boldsymbol{d}},\boldsymbol{M[a,b]}\in[\boldsymbol{0},\boldsymbol{1}]$$

其中 $\boldsymbol{S(a-b)}$ 为向量 $\boldsymbol{(a-b)}$ 中包含相同元素的个数，$\boldsymbol{d}$为 $\boldsymbol{a}$、$\boldsymbol{b}$的维度。

式（2）衡量的是两乐汇形成模进的规模。当 $\boldsymbol{M(a,b)=1}$ 时，表示两乐汇为严格模进（也包括完全重复）；当$\boldsymbol{M(a,b)=0}$, 表示两乐汇无相似度。

于是，可以定义一个相似度概率分布范围来定义两乐汇是否有相似关系，在本研究中我们将这个值设定为 50%，如式（3）。

$$\boldsymbol{M(a,b)\geq 0.5}$$

当然，这个方法也适用于带有节奏的音乐序列，而不仅仅是音高间的关系。判定方法与式（2）（3）相同。

3.4 LSTM

LSTM[16](Hizlisoy, S., Yildirim, S., & Tufekci, Z.，2021) 接收三个输入：数据输入 X（音乐序列），以及来自上一个单元的输出h_{t-1}和 C_{t-1}。LSTM 处理数据的流程如下图所示：

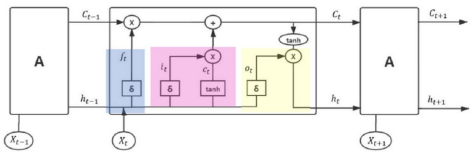

图 2. 一个 LSTM 单元的计算过程

LSTM 由三个门组成：如（图 2）所示，其中蓝色的为遗忘门，粉色的为输入门，黄色的为输出门。

遗忘门的作用是选择保留多少从上一个单元输入的信息，其计算如式（3）：

$$\boldsymbol{f_t}=\boldsymbol{\delta}(\boldsymbol{W_f}[\boldsymbol{h_{t-1}},\boldsymbol{X_t}]+\boldsymbol{b_f})$$

其中 h_{t-1}式上一个时刻的隐藏输出，X_t 当前时刻的输入，W_f、b_f 是可学习的权重与偏置，δ是 Sigmoid 激活函数。

输入门的计算包含两个部分 i_t、c_t 两个部分，最终将 i_t 、c_t 相乘后输出，i_t、c_t 的计算方法如式（4）式（5）所示：

$$\boldsymbol{i_t}=\boldsymbol{\delta}(\boldsymbol{W_i}[\boldsymbol{h_{t-1}},\boldsymbol{X_t}]+\boldsymbol{b_i})$$

$$\boldsymbol{c_t}=\boldsymbol{\tanh}(\boldsymbol{W_c}[\boldsymbol{h_{t-1}},\boldsymbol{X_t}]+\boldsymbol{b_c})$$

其中，W_i、W_c、b_i 、b_c均为可学习参数。

输出门负责计算输出的 h_t，其计算方法如式（6）（7）：

$$\boldsymbol{o_t}=\boldsymbol{\delta}(\boldsymbol{W_0}[\boldsymbol{h_{t-1}},\boldsymbol{X_t}]+\boldsymbol{b_0})$$

$$\boldsymbol{h_t}=\boldsymbol{o_t tanh}(\boldsymbol{C_t})$$

最后，输出 C_t 的计算方式如式（8）：

$$\boldsymbol{C_t}=\boldsymbol{f_t c_{t-1}}+\boldsymbol{i_t c_t}$$

以上是一个 LSTM 单元的全部计算过程。在本文中，我们通过使用多层堆叠的形式来生成音乐。首先，数据通过 LSTM 层，然后经过 Dropout 丢弃一部分信息，再进入第二个 LSTM 层，再经过 Dropout 丢弃一部分信息，然后进入第三个 LSTM 层，随后经过一个全联接层提取特征，再通过 Dropout 丢弃一部分信息，最后再经过一个全连接层后进入 Softmax 运算（9），预测下一个X 的概率分布并取最概率以生成音乐（10）。

$$\boldsymbol{P(h_t)}=\frac{\boldsymbol{e^{h_t}}}{\sum_{\boldsymbol{j=1}}^{\boldsymbol{N}}\boldsymbol{e^{h_t}}}$$

$$\boldsymbol{y_t}=\boldsymbol{argmax P(h_t)}$$

由上述 LSTM 生成音乐的过程可以看出，LSTM 的输入依赖上一个时刻的输出，因此其能够很好的抽象数据在时序上的关系。但是同时也暴露出 LSTM 的缺点：其无法并行计算，只能按照时间顺序逐个计算。还有，LSTM 虽然减缓了梯度消失的问题，但如果序列太长，反向传播算法中的梯度在网络的较深层中逐渐变得非常小，以至于权重更新几乎没有效果，导致网络难以学习到深层次的特征和关联。这是因为在反向传播过程中，每一层都要计算梯度以调整权重，而每一层的梯度都是由上一层传递下来的，如果梯度值接近于零，多次相乘后将导致整体的梯度非常接近于零，从而无法有效地更新权重。

3.5 Transformer

Transformer（Vaswani, A., Shazeer, N., Parmar. 等，2017) 最初是为机器翻译设计的模型，它引入了一种全新的编码器 - 解码器架构（Encoder-Decoder Architecture），用于处理序列数据，如自然语言文本、音乐等。由于这种编码器 - 解码器架构不依赖于 RNN 和 CNN，只依靠注意力来建立数据之间的联系，因此其有效地避免了梯度消失与梯度爆炸的问题。

此外，它可以通过将位置标记为一个位置编码，因此 Transformer 可以做大规模的并行计算，而无需按顺序逐个处理。

自注意力机制允许模型同时考虑输入序列中的所有位置，并根据它们的重要性来分配权重。这使得模型可以更好地捕捉序列中的长距离依赖关系。在 Transformer 模型中，Q（Query）、K（Key）和 V（Value）是自注意力机制（Self-Attention）的关键组成部分，它们之间的关系如（11）所示：

$$\mathrm{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\top}}{\sqrt{\boldsymbol{d}}}\right)\boldsymbol{V}$$

用于音乐生成的 Museformer 则在此基础上做了改进，它提出了一种叫做“粗粒度”、“细粒度”的注意力机制[11]。其接收两个顺序输入 $\boldsymbol{X}$ 与 $\boldsymbol{X'}$，其中 $\boldsymbol{X}$ 为源（一段音乐序列）与 $\boldsymbol{X'}$ 为目标（继续生成的音乐序列），d 为嵌入维度。它计算 $\boldsymbol{X'}$ 中的每个元素的可能性概率分布，如式（12）

$$\mathrm{Attention}(\boldsymbol{x_i'},\boldsymbol{X}) = \mathrm{softmax}\left(\frac{\boldsymbol{x_i'^{T}}\boldsymbol{W_Q}(\boldsymbol{X}\boldsymbol{W_K})^{T}}{\sqrt{\boldsymbol{d}}}\right)\boldsymbol{X}\boldsymbol{W_V}$$

此外，它还通过汇总每个小节的摘要信息来捕捉小节之间的相关度，从而生成符合音乐结构的乐曲。

给定第 i 小节，我们令其摘要标记为 $s_i \in R^{1\times d}$，给定音乐序列 $X_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,|x_i|}] \in R^{|x_i|\times d}$，则摘要标记的计算如（13）：

$$\tilde{\mathrm{s}}_{\mathrm{i}} = \mathrm{Attn}(\mathrm{s}_{\mathrm{i}}, [\mathrm{X}_{\mathrm{i}}, \mathrm{s}_{\mathrm{i}}])$$

最后，汇总属于与结构相关的小节和当前小节内的标记的信息，以及其他小节的摘要信息，然后更新概率分布。更新表示如 (14)：

$$\tilde{x}_{i,j} = \mathrm{Attn}\left(x_{i,j}, \left[X_{R(i)}, X_{i,k\le j}, \tilde{S}_{\bar{R}(i)}\right]\right)$$

在该公式中，$\boldsymbol{x_{i,j}}$ 只与属于与结构相关的小节和当前小节内的前面标记相关，对于其他小节，它只与它们的摘要标记相关。

3.6 评价

交叉熵在序列生成类任务中的应用非常广泛，它是一种常用的损失函数，通常用于训练语言模型等。在训练过程中，模型的目标是最小化交叉熵，从而更新参数，这意味着模型在生成文本或音乐时会更接近训练数据的分布，从而提高生成的质量和准确性。因此，生成类任务的交叉熵是一个重要的评估指标，用于指导模型的训练和改进。

$$L(y, y') = -\sum_{x} y \log(y')$$

困惑度用于衡量能否正确预测下一个序列，它是一个大于等于 1 的值。数值越接近 1，代表模型的性能越好。如果困惑度值等于 1，则说明其完美地预测了下一个序列。

$$\text{Perplexity} = e^{-\frac{1}{N}\sum_{i=1}^{N} log_e P(x_i|x_1, x_2, \ldots, x_{i-1})}$$

音乐艺术是感性与理性的结合，因此主观评估也是音乐评价不可缺少的一个环节。人工评价以问卷的形式进行，共 10 人。我们设置 20 首曲目，其中 5 首为 LSTM 生成，5 首为 Transformer（Museformer）生成，然后让听众随机挑选 5 首他觉得符合人类音乐创作习惯的乐曲，满分 5 分。模型得分越高，证明其创作的音乐越接近人类创作音乐的听感。

4.实验

4.1 数据集

我们使用了 1000 首流行音乐的旋律 MIDI 文件作为数据集，这些文件是通过人工听写近五年的流行歌曲所记录下来的，主要风格有 Pop 抒情、流行摇滚、以及中国风、古风歌曲，总时长共计 56 个小时。我们将这些 MIDI 文件随机以 8:1:1 的比例用作训练、验证、测试集。

4.2 实验设置

我们在 Tesla T4 GPU 上对 Museformer 与 LSTM 以相同的数据集以及划分方式分别进行实验。其中：

Museformer 以 Pytorch 框架实现，层数为 4，隐藏大小 512，注意力头数为 8，前馈神经网络的隐藏大小为 2048，基本保持了 Museformer 的原配置。训练时的学习率设置为 0.0001，批量大小设置为 32，采用 Adam 优化器，模型经过 10000 次迭代。

LSTM 的输入需要以独热向量表示。在网络中，我们使用了 3 个 LSTM 层，隐藏层中神经元的个数为 1024，3 个 Dropout 层，Dropout 丢弃系数为 0.7，此外在其中链接了两个 Dense 层。模型的迭代次数为 10,000 次。

4.3 比较

实验结果表明，Transformer 的表现优于 LSTM。

首先，我们分别统计了序列长度为 512、1024、2048、4096 的困惑度值，如表 1 所示：

表 1：不同表示方法在不同长度上的困惑度值对比

	PPL（512）	PPL（1024）	PPL（4096）	SE
LSTM	3.33	4.98	6.86	9.92%
Museformer	1.53	1.77	2.73	5.88%

如图可以看出，序列长度较小时，Transformer 优于 LSTM，但两者差距不大。但序列长度增加至 4096 时，两者困惑度值的差距开始变大。这也说明 LSTM 对于处理长序列任务而产生的梯度问题仍然是其限制。

其次，我们通过现场问卷的形式进行主观评价。我们选取 10 人作为调查对象，随机设置 20 首曲目，其中 5 首为 LSTM 生成，5 首为 Transformer（Museformer）生成，然后让听众随机挑选 5 首他觉得符合人类音乐创作习惯的乐曲，满分 5 分，共测试三轮，每轮测试后对分数取均值。其统计结果如下：

表 2：Transformer 与 LSTM 的主观评价得分

	第一轮	第二轮	第三轮	均值
LSTM	2.6	3.1	2.8	2.83
Museformer	2.5	2.4	3.7	2.86

可以看出，音乐在人性化方面差距并不大，表现出了实力相当的趋势，这说明两者在处理音乐性方面并无差距。但侧面也说明了两者生成的音乐距离人类创作的音乐还有一定的差距。

5.结论

本文通过 Tranformer 与 LSTM 模型在音乐生成任务上的对比，分析了其各自的优缺点。但值得肯定的是，Transformer 模型仍然在性能上优于 LSTM。并且，Transformer 模型能够大规模并行计算的能力能够使其拥有更加广阔的前景。但是，两者生成的音乐旋律仍然与人类创作的音乐有一定的距离，这也是该领域当下正在解决的问题。不过，深度学习的发展总是超乎我们的想象，未来的前景仍然非常广阔。Transformer 在音乐作曲上的巨大潜力使得越来越多的人们投身于该领域的研究。当然，即使 Transformer 表现出色，但 LSTM 也不可替代。我们只能期待未来的研究能够解决两者现有的问题，让计算器创作出更加符合音乐形式、更加符合人类审美的音乐。

参考文献

Devlin, J., Chang, M.-W., Lee, K., et al. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171 – 4186).

Dai, Z., Yang, Z., Yang, Y., et al. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. In Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL) (pp. 2978 – 2988).

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Simon, I., et al. (2018). Music transformer: Generating music with long-term structure. In Proceedings of International Conference on Learning Representations (ICLR).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735 – 1780.

Katharopoulos, A., Vyas, A., Pappas, N., et al. (2020). Transformers are RNNs: Fast autoregressive transformers with linear attention. In Proceedings of International Conference on Machine Learning (ICML) (pp. 5156 – 5165).

Yu, B., Lu, P., Wang, R., et al. (2022). Museformer: Transformer with Fine- and Coarse-Grained Attention for Music Generation. In Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS).

Rae, J. W., Potapenko, A., Jayakumar, S. M., et al. (2020). Compressive transformers for long-range sequence modelling. In Proceedings of International Conference on Learning Representations (ICLR).

Rizvi, D. R., Nissar, I., Masood, S., et al. (2020). An LSTM based deep learning model for voice-based detection of Parkinson's disease. International Journal of Advanced Science and Technology, 29(8).

Roy, M., Saffar, A., Vaswani, A., et al. (2021). Efficient content-based sparse attention with routing transformers. Transactions of the Association for Computational Linguistics (TACL), 9, 53 – 68.

Shaw, P., Uszkoreit, J., & Vaswani, A. (2018). Self-attention with relative position representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers) (pp. 464 – 468).

Vaswani, A., Shazeer, N., Parmar, et al. (2017). Attention is All You Need. In Advances in Neural Information Processing Systems.

Wu, F., Wu, T., Qi, B., Jiao, D., Jiang, D., et al. (2021). Smart bird: Learnable sparse attention for efficient and effective transformer. arXiv preprint arXiv:2108.09193.

Hsiao, W.-Y., Liu, J.-Y., Yeh, Y.-C., et al. (2021). Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 35, 178 – 186.

Zhu, H., Chen, E., Liu, Q., Yuan, N., et al. (2018). Band: A melody and arrangement generation framework for pop music. In The 24th ACM SIGKDD International Conference (pp. 2837 – 2846).

Zeng, M., Tan, X., Wang, R., et al. (2021). MusicBERT: Symbolic music understanding with large-scale pre-training. In Proceedings of Findings of the Association for Computational Linguistics (ACL Findings) (pp. 791 – 800).

Hizlisoy, S., Yildirim, S., & Tufekci, Z. (2021). Music emotion recognition using convolutional long-short-term memory deep neural networks. Engineering Science and Technology, an International Journal, 24(3), 760 – 767.