arXiv:2301.01162v1 [cs.SD] 3 Jan 2023

语言模型是鼓手：利用自然语言预训练进行鼓乐创作

Li Zhang, Chris Callison-Burch

宾夕法尼亚大学{z
harry,ccb}@seas.upenn.edu

摘要

利用人工智能进行自动音乐生成通常需要大量数据，而对于许多不常见的音乐流派和乐器而言，这类数据很难获取。为解决这一问题，我们介绍了正在进行的研究及初步发现，探讨深度模型通过迁移学习将知识从语言迁移到音乐的可能性：仅使用数百个鼓演奏 MIDI 文件，对在海量文本语料库上预训练的大语言模型进行微调。我们表明，通过这种方式，最大的前沿模型之一 GPT-3 能够生成合理的鼓律动；而未经预训练的模型 Transformer 除了简单重复之外，不具备这种能力。评估生成的音乐是一项具有挑战性的任务，而由于相关文献中的先例很少，评估鼓律动则更加困难。因此，我们提出了一种定制的结构化评估方法，并分析 GPT-3 生成的鼓律动与人类专业鼓手演奏的鼓律动，从而揭示这种语言到音乐迁移生成方式的优势与不足。我们的研究结果表明，利用大语言模型进行语言到音乐的迁移学习是可行且前景广阔的。[1][2]

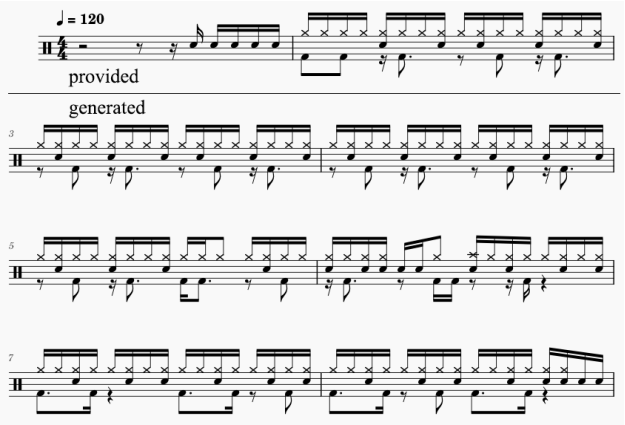

图 1：我们生成的鼓轨示例乐谱，展示了模型在一定程度上遵循动机并进行变化的音乐能力。前两个小节为给定内容，其余部分由模型生成。

引言

使用人工智能理解和生成音乐有着悠久的历史（Roads 1985），近年来也持续受到关注（Kaliakatsos-Papakostas、Floros 和 Vrahatis，2020）。一类研究关注的是符号音乐而非音频，即将音乐表示为 MIDI 等序列数据。尽管音乐与语言之间的类比早已得到研究（McMullen 和 Saffran，2004），但音乐的符号表示在表层形式上与语言表现出更明显的相似性。例如，音乐包含音符、小节和乐段，而语言包含词元、句子和段落。因此，将自然语言处理（NLP）技术应用于音乐是顺理成章的。具体而言，大多数研究都试图学习音乐的嵌入空间（Liang 等，2020），使之类似于文本的嵌入空间。

近期研究已将 Transformer（Vaswani 等人（2017））用于符号音乐处理（Huang 等，2018；Yu 等人（2022））。Transformer 在海量文本语料库上进行预训练后，被称为大语言模型（LLM），是当前许多自然语言处理任务中最先进的方法（De- vlin 等，2019；Brown 等，2020）。值得注意的是，Zeng 等人（2021）的研究是最早、也是仅有的几项研究之一：该研究在包含超过 100 万首歌曲的大型符号音乐语料库上对 Transformer 进行预训练，采用了与在文本语料库上进行预训练类似的方法，并在各种音乐理解任务中取得了最先进的性能。然而，这项研究以及大多数其他数据驱动的音乐处理研究最显著的局限之一，是训练所需的符号音乐数据量很大；对于非主流音乐流派、特定风格、较不常见的乐器或非同寻常的规格而言，这类数据极难获取，从而严重限制了此类方法的通用性。因此，低资源符号音乐处理仍然极具挑战性。另一方面，对于文本而言，由于大语言模型的预训练文本数据规模极其庞大，远超音乐数据，少样本学习得到了大语言模型的大力推动。

在这项工作中，我们率先探索这种**文本-**

版权所有 © 2023年，人工智能促进协会（www.aaai.org）。保留所有权利。[1]已被 AAAI 2023 创意人工智能跨模态研讨会接收。[2]数据和代码见 https://github.com/zharry29/drums-with-llm。本文标题戏仿了自然语言处理领域中将论文标题写成“语言模型是……”的流行风潮。

到音乐的迁移学习潜力在大语言模型中。换言之，我们提出假设：如今最先进的大语言模型经过海量文本数据的预训练后，能够在**几乎不使用音乐数据的情况下生成符号音乐**，并达到一定程度。我们特别聚焦于一种乐器——架子鼓，原因有多方面。首先，架子鼓是爵士乐、放克、布鲁斯、福音音乐、拉丁音乐、流行乐、摇滚乐、金属乐等许多音乐流派中最常见且最重要的乐器之一。其次，架子鼓的符号表示比大多数有音高乐器更简单，因为每个音符都没有音高，而是对应于敲击某一面鼓。由于鼓的数量通常远少于可能的音高数，因此生成的序列要短得多，也更容易由模型处理。第三，与许多其他乐器相比，架子鼓演奏通常在听众审美方面具有更大的自由度，因此适合作为研究大语言模型音乐生成的切入点；人们普遍认为，这是一项极具挑战性的任务。

我们聚焦于鼓乐生成或创作这一任务，而相关文献中的已发表研究数量较少。尽管现有研究几乎无一例外地将鼓作为伴奏处理，但我们转而关注鼓独奏生成，这与故事生成有着便利的类比，而大语言模型已知在故事生成方面表现出色。

我们在 Groove 数据集（Gillick等人（2019））上对最先进的大语言模型 GPT-3 模型（Brown 等，2020）进行微调。该数据集包含约 400 段以 MIDI 录制的鼓律动演奏。为了利用 GPT-3 的文本预训练能力，我们提出了一种鼓乐演奏的文本表示。我们提出以下核心发现：1. 最大和最小的 GPT-3 模型在微调后都能够生成具有实质内容的鼓律动。2. 然而，一个规模相近但未接受语言数据预训练的模型无法做到这一点。

我们认为，现有的音乐生成自动评估方法不足以应对我们的任务。因此，我们针对鼓律动提出了一套评估方法，从定性和定量两个方面评估机器生成的鼓律动相较于人类演奏的优势与不足。最后，我们提供了一些初步的聆听测试结果，并计划在未来的工作中开展规模更大且更为严谨的测试。

架子鼓

架子鼓（也称为 drum kit，口语中也常简称为鼓）是一种由多个子乐器组成的复合乐器，包括鼓和镲片；在本文中，两者通常都统称为鼓。这里，我们假设一种简单、典型的架子鼓配置，包括踩镲、碎音镲、叮叮镲、底鼓、小军鼓和一只通通鼓（见图 2，但不包括高音通通鼓和落地通通鼓[3]）。

架子鼓的演奏可以记谱为乐谱（一种人类可读的符号表示），也可以记为 MIDI；MIDI 记录了每次击打鼓的时间和力度，是一种计算机可读的符号表示。

[3]https://i.redd.it/cen72phsihb81.png

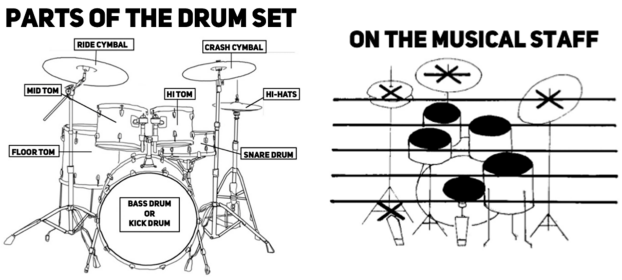

图 2：基础架子鼓的构造，以及乐谱中各个鼓的呈现方式。

	训练	dev	test
MIDI 总数	373	47	35
- 摇滚乐	169	16	15
- 爵士乐	41	6	4
- 拉丁音乐	37	10	3
- 放克	31	6	4
- 嘻哈	26	1	3
- 其他	69	8	6

表 1：本研究使用的筛选后 Groove 数据集中各风格的 MIDI 文件数量。

数据集

在为数不多的鼓乐演奏数据集中，谷歌的 Groove MIDI 数据集[4] 是迄今为止规模最大、质量最高的数据集，包含 1,150 个 MIDI 文件，以及 10 位专业鼓手演奏的超过 22,000 小节击鼓内容。在该数据集中，鼓乐演奏分为两类：由节奏构想组成的长序列，即节奏型；以及短暂、自由流淌的表达，即加花。由于我们关注的是长序列中的鼓乐生成或作曲，因此只考虑数据集中的节奏型。每个 MIDI 均标注了风格（例如摇滚乐、放克、福音音乐等）、速度（以每分钟拍数，即 BPM，表示）和拍号。为求简便，我们只考虑四四拍。我们沿用数据集中的训练集-开发集-测试集划分。筛选后的 Groove 数据集统计信息如图 1 所示。

Groove 数据集最初旨在研究微观时序和表现性演奏，因此其中的架子鼓 MIDI 文件包含了人类演奏的不完美性。然而，我们将该数据集重新用于研究鼓乐创作，因此做出以下简化。首先，我们将所有音符量化到十六分音符网格上。换言之，MIDI 中的所有音符事件都会重新定时到每小节 16 个等间隔时间戳中最近的一个。这样量化意味着有意的非网格演奏，例如三连音或摇摆律动，会丢失。其次，我们舍弃力度信息（即击打鼓的力度），而这通常可以在事后以粗略的精度推断出来。第三，虽然架子鼓可以通过多种表现方式和奏法演奏（例如使用不同工具的不同部位击打鼓的不同部位），但我们将其简化为基本的

[4]https://magenta.tensorflow.org/datasets/groove

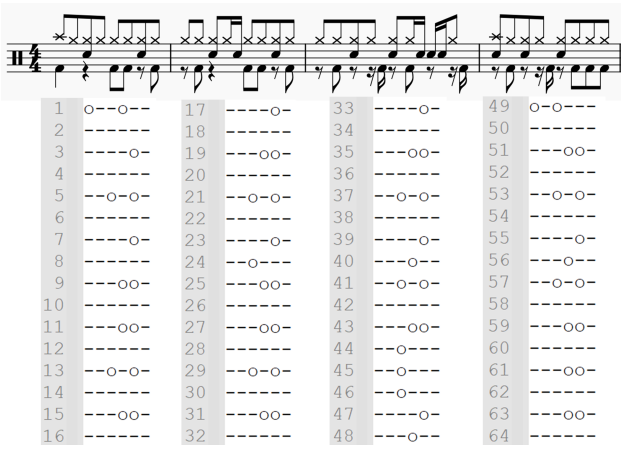

图 3：鼓谱示例的鼓点滚动表示。每小节在文本中对应 16 行，每行对应一个十六分音符。每行包含 6 个字符，分别对应 6 个鼓，其中“o”和“-”表示相应的鼓是否被敲击。

踩镲、碎音镲、叠音镲、底鼓、军鼓和落地通鼓的奏法（击打鼓面）。换言之，每个音符只有 6 种可能取值。第四，我们将每个 MIDI 文件截取为仅保留前 16 小节；例如，在每分钟 128 拍的速度下，这相当于 30 秒。最后，我们移除开头第一个四分音符为休止符的空小节，并忽略不足 8 小节的节奏型。

实验

表示

如前所述，大语言模型展现出了极强的小样本迁移学习能力，能够将一种文本任务中的知识迁移到其他任务中。为了探索如何将这一能力应用于音乐，我们需要为鼓律动设计一种文本表示。我们提出了一种类似钢琴卷帘的表示（Brunner 等，2018），称为鼓点滚动，它本质上是一个多行字符串，其中每一行对应时间序列中的一个十六分音符，而每行中的每个字符表示是否敲击某个鼓。具体而言，如果在某个十六分音符处敲击了相应的鼓，则用“o”表示；否则用“-”表示。示例见图 3。为了帮助大语言模型识别小节边界，我们在每 16 行（即一个小节）后添加一个内容为“SEP”的换行，并在最后一行后添加一个内容为“END”的换行。

Task

我们聚焦于鼓乐生成中的一种任务，称为鼓点补全：模型获得前 2 小节，并须补全该节奏型剩余的 14 小节。模型可在任意位置终止。这类似于自然语言处理中基于条件的故事生成。

模型

首先，我们考虑两种简单的基线：**随机**选择是否演奏一个音符，以及**重复**给定的第二个小节。随后，我们在训练集上微调一个最先进的大语言模型——拥有 1750 亿个参数的 OpenAI**GPT3 Davinci** 。在每个鼓律动文件中，输入（提示）是前 2 小节，输出（补全）是剩余的 14 小节。温度设为 0.85，以鼓励创造力。使用 OpenAI 的 API 在训练集上微调该模型，花费 38.33 美元，耗时约 30 分钟。

为了确定模型规模在鼓乐生成中所起的作用，我们还考察了一个规模较小的 **GPT-3 Ada** 模型，其参数量为 3.5 亿。该模型在相同的语料库上进行过预训练，并采用与更大的 Davinci 模型相同的设置。使用 OpenAI 的 API 在训练集上微调该模型，费用为 0.51 美元，耗时约 5 分钟。后来发现，这两个 GPT-3 模型都能够生成有意义的鼓律动。

为了确定语言预训练在鼓乐生成中所起的作用，我们将对照组设为未经预训练的 GPT-3 模型，即一个与 GPT-3 规模相同的 Transformer（Vaswani 等人（2017））。由于我们的计算资源无法支持像 Davinci 那样大的模型，而且训练集仅有 373 个文本文件、每个文件 256 行，这很可能不足以使训练损失收敛，因此我们对一个规模较小、未经预训练的 Transformer 进行微调，其参数量为 8500 万，与 GPT-3 Ada 的规模相当。尽管训练损失确实收敛了，但无论输入什么 2 小节，该模型都会预测出相同的特定序列，表现不比随机基线更好。这表明，语言预训练是有效进行鼓乐生成的必要条件。

评估

GPT-3 生成的鼓律动有多好？按照音乐生成领域的惯例，我们同时进行客观／自动评估和主观／人工评估。我们报告了基于测试集得出的发现。

客观评估

我们认为，符号音乐生成领域已有的自动评估方法不适用于我们的任务。举几个例子：例如，研究早已表明，困惑度与自然语言处理中的人类感知相关性较低（Kuribayashi 等人（2021））；结构相似度虽然是合理的指标，但往往通过一些简单的相似度度量来近似（Yu 等人（2022）），而且会抑制偏离参考的创造性，而参考绝不应被视为真实值。

相较之下，我们探讨了何为优秀的鼓律动，并提出一种称为**节奏型与加花分析**的结构评估方法。我们假设，优秀的鼓律动通常至少满足以下标准：

1. 存在一个或多个体现某种节奏构想且具有一致性的**节奏型**，并偶尔出现称为**加花**的变化。 2. 节奏型中的小节应足够相似，但理想情况下并非完全相同。 3. 加花中的小节应与相邻节奏型中的小节有足够大的差异。

随后，我们验证人工演奏的律动基本符合这些标准。

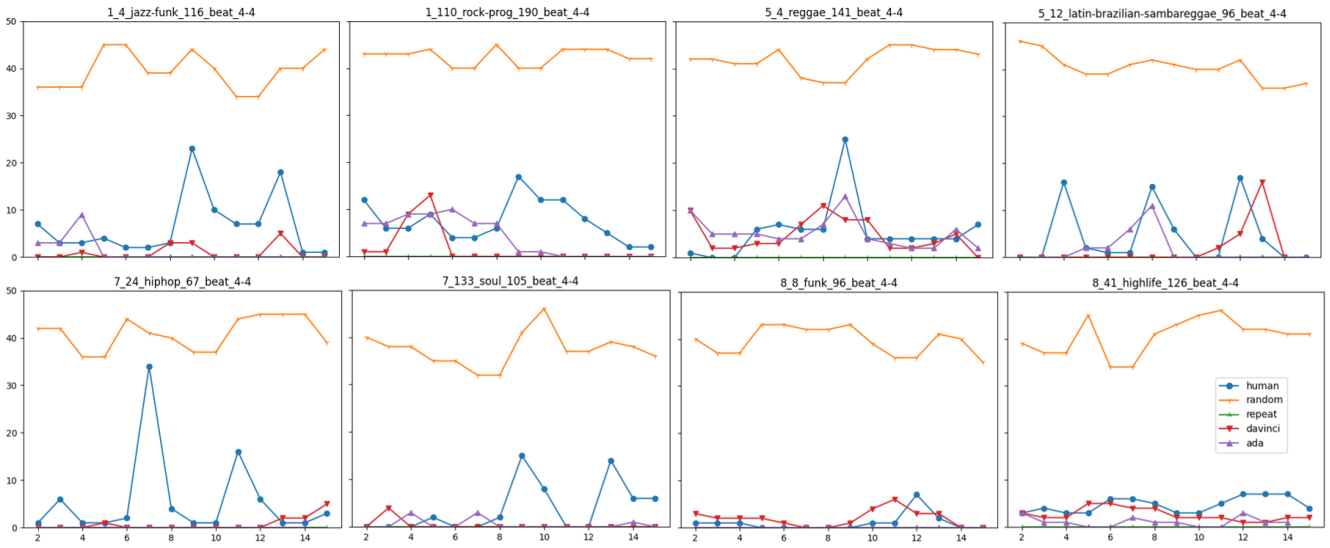

图 4：变　　化（纵轴）表示每个小节（横轴）的变化，图中展示了从每种风格中随机抽取的 8 个鼓律动　　.

	人类	随机	重复	Davinci	Ada
平均变化	5.1	40.4	0	3.0	3.4
平均内部质心	10.1	5.7	0	6.9	7.2
平均之间质心	1.4	1.2	0	0.6	0.8

表2：通过不同方式生成的鼓点节奏型的定量统计。聚类质心之间的距离表明节奏型和加花的差异程度。聚类质心与小节之间的距离表明同一类别中各小节的变化幅度。

	人类	随机	重复	Davinci	Ada
重复性	0	0	35	3	7
一致性	32	0	0	29	15
混乱	3	35	0	3	13
有加花	30	0	0	13	10
平均长度	13.3	16	16	13.7	12.5

表 3：测试集中各模型生成的鼓律动中，被判定符合各项标准的数量。

在以滚奏表示的鼓乐演奏中，每个小节都由 16 行字符串表示（图 3）。为了将每个小节分类为节奏型或加花，我们取一个大小为 3、以某个小节 m_i 为中心的滑动窗口，并计算该小节与其两个相邻小节之间的编辑距离。这两个距离中的较小值称为中心小节的变化度：

$$variation(m_i) = \min(dist(m_i, m_{i-1}), dist(m_i, m_{i+1}))$$

直观来说，节奏型中一个小节的变化度应较小，而加花中的变化度则应较大。因此，在良好的鼓律动中，除第一个和最后一个小节外，所有小节的变化度都应有明显区分。

接下来，我们依次绘制开发集里每种鼓律动中各小节的变化度，并观察到了一致的模式。图 4 展示了随机选取的八种不同风格的律动。直观来说，对于人类演奏，节奏型中某个小节的变化度应该较小，而加花中的变化度应该较大。大多数示例确实如此：人类演奏的节奏型具有一致性，但也存在一些变化（平台）；而加花则有很大不同（尖峰）。正如预期，随机基线会导致所有小节的变化度都很高，而重复基线则完全没有变化——两者都不可取。

定性考察表明，不同规模的两个 GPT-3 模型显然都能生成颇具内容的鼓律动，其中有许多平台和偶尔出现的尖峰，不过尖峰比人工演奏的少。定量而言，我们计算了所有律动中所有小节的平均变化度。如表 2 所示，GPT-3 模型生成的鼓律动往往比人工演奏的变化度更小。为了观察生成的节奏型是否变化度较小，而加花的变化度是否大得多，我们使用 K-means 聚类，根据变化度将一个鼓律动中的小节分为两类。随后，我们计算两个质心之间的平均类内距离，以及每个小节与其所属质心之间的平均类间距离。如表 2 所示，质心间距离表明，与 GPT-3 和随机基线相比，人类演奏的律动中，节奏型与加花的区分更加鲜明。质心间距离还表明，与 GPT-3 生成的律动相比，人工演奏的律动中，节奏型或加花类别内部的变化度分布更为显著。

主观评估

我们的客观评估显然能提供有用信息，但仍不充分。因此，我们进行了聆听测试，并依据以下标准进行错误分析：

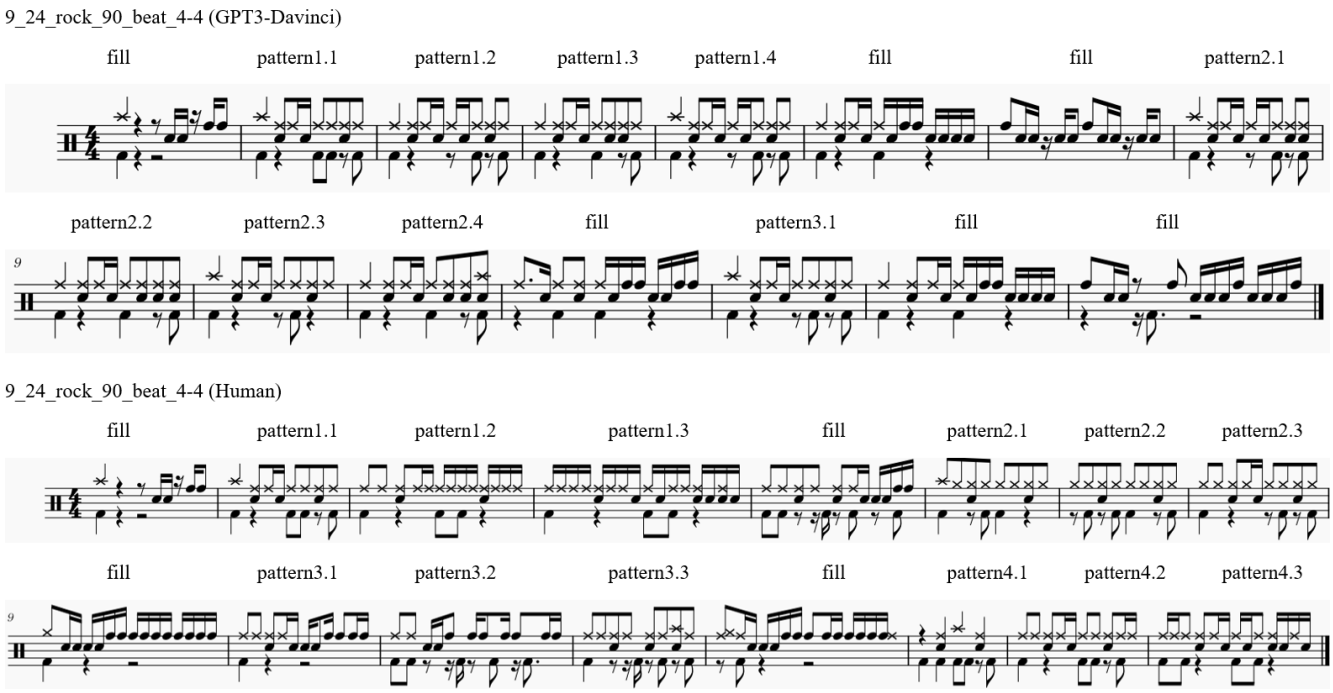

图5：GPT-3 生成的一个令人满意的鼓律动乐谱（上），与数据集中人类演奏的、前两个小节相同的鼓律动（下）并列展示。我们将每个小节标注为节奏型或加花。例如，节奏型$x.a$ 和节奏型$x.b$ 相邻，具有相同的重音后拍，但并不完全相同。

基于以下标准进行错误分析：

- 鼓律动是否具有重复性，也就是说，各小节之间几乎没有或完全没有变化度?
- 鼓律动是否具有一致性，也就是说，各小节之间存在一定变化度，但仍能跟随一个稳定的节奏构想（具体而言，即后拍的位置）？
- 鼓律动是否混乱，也就是说，变化度过大或缺乏明确的节奏构想?
- 鼓律动是否包含合理的鼓加花?

虽然以严谨方式扩大这项分析并精心挑选受试者的工作留待未来开展，但我们在表 2 中列出了自己的判断，作为初步发现。具体而言，我们将以不同方式创作的所有鼓律动打乱顺序，然后随机呈现给一位作者评判；这位作者有多年的击鼓训练经历。自然，所有随机生成的律动都被评为混乱，缺乏一致性的动机，而所有重复的律动本质上都具有重复性。对于人类演奏的律动，大多数被评为具有一致性，而且大多数至少包含一个加花，其变化足以区别于其余节奏型。相比之下，GPT3 Davinci 生成的律动在一致性方面仅略逊一筹，各小节之间也有令人满意的变化度，但其中包含加花的律动显著较少，因此整体上不够有趣，也更容易预测。对于规模较小的 GPT-3 Ada 模型，这一观察更为明显：其生成的律动一致性更差，加花也更少。

案例研究与启示

我们认为，大语言模型仅凭数百个训练样例，在不了解应关注哪些特征的情况下，就展现出了创作优质鼓律动的惊人能力。从上述客观评估和主观评估中，我们观察到，大语言模型和人类都能够创作出具有结构和一定变化度的鼓律动。然而，模型的鼓乐创作为何仍不如人类？我们提出了几个可能的因素。

我们发现，机器生成的鼓律动缺乏音乐性的一个常见原因，是后拍位置不当。后拍是每小节中稳定重音的节拍，在许多音乐风格的四四拍中，通常位于第 2 拍和第 4 拍。人类鼓手演奏变化时，往往会遵循后拍的位置，而模型则倾向于忽略这一概念。缺乏音乐性的另一个原因是缺少鼓加花。人类鼓手通常会自然而然地考虑节奏型与加花之间的循环，以带来引人入胜的演奏，而模型尚未认识到这些偶尔偏离的重要性。

为了定性检验这些观点，我们展示了两组由 GPT-3 创作的鼓律动，一组展现其优势，另一组展现其不足；同时还展示了采用相同前两个小节的人类演奏鼓律动。

图5展示了一个正面示例。显然，人类和 GPT-3 创作的两个律动都包含一致的节奏型，并穿插着加花，这是大多数听众所期待和喜爱的典型结构。仔细观察节奏型的分组，可以发现两个律动都改变了底鼓和军鼓的落点——这称为切分音，同时仍将后拍稳稳地保持在第2和第4个四分音符上。

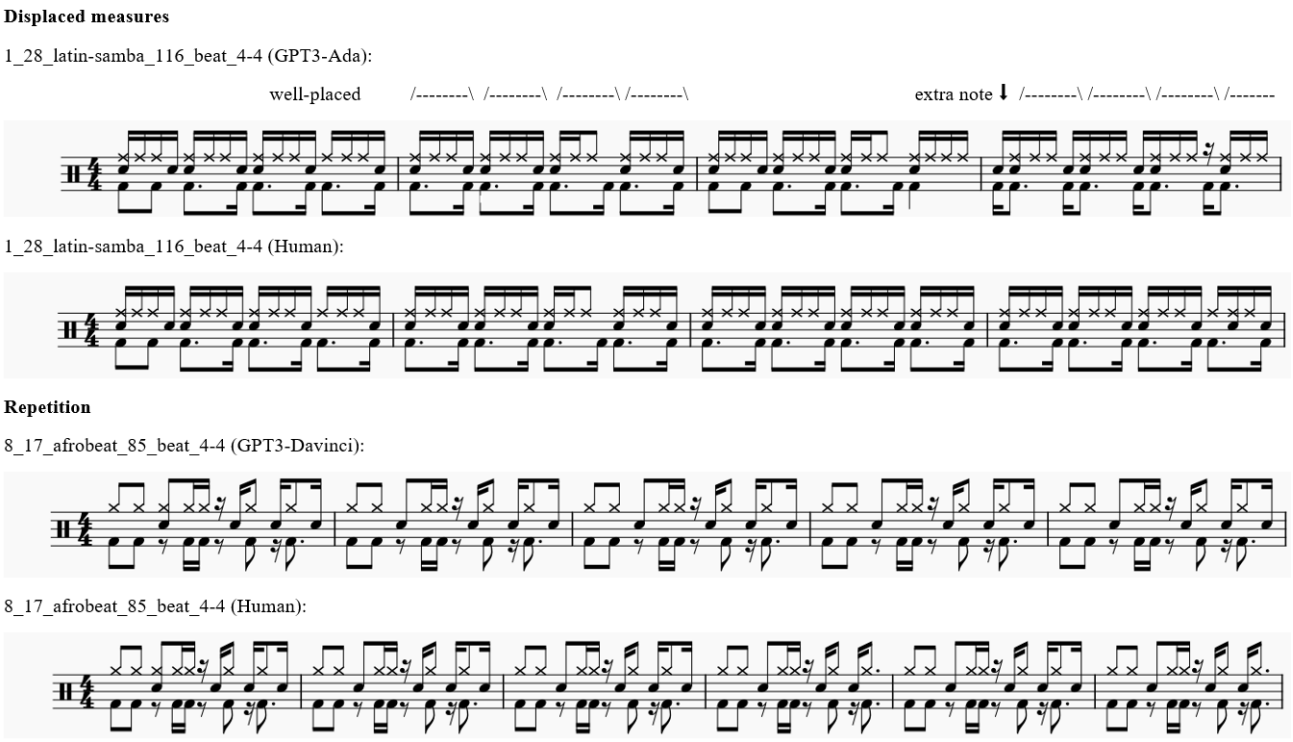

图 6：GPT-3 生成的两个有问题的鼓律动乐谱（第 1 个和第 3 个），与人类演奏的乐谱（第 2 个和第 4 个）并列展示。第一种情况下，GPT-3 在第 4 小节开头多生成了一个音符，导致之后所有节奏型都发生错位。第二种情况下，GPT-3 生成了严格重复的小节；休止符被省略。

作为典型的鼓律动，两个律动中的节奏型都始终包含军鼓击打（人类节奏型3.1和3.2中有两个例外）。这种作曲方式非常理想，因为稳定的后拍位置以扎实的节奏基础引导听众，而变化则让律动保持趣味性和自然感。除了能够遵循人类鼓手的标准演奏习惯外，这个示例还展现了大语言模型的创造力。人类演奏的律动严格遵循了“3小节节奏型后接1小节加花”的模板，而 GPT-3 生成的律动则不然。它采用了跨越多个小节的加花，更有意思的是，还将奇数个节奏型编为一组，随后接上加花；这种做法在该音乐流派中并不常见。

图 6 展示了两个负面示例，代表 GPT3 生成鼓律动时的两个常见问题。第一个案例称为小节错位。人类鼓手演奏节奏型时，通常会为每个小节安排一些节奏构想。换句话说，两个小节之间的界限通常很清晰。在我们的鼓点滚动表示中，每个小节由 16 行文本组成，后面跟着一个换行符“SEP”，以标示这种界限。然而，GPT-3 Ada 有时无法遵守这些界限，导致小节错位，因此生成了表 3 中所述的更加“混乱”的节奏型。相比之下，GPT3 Davinci 犯这类错误的情况要少得多。第二个案例展示了大语言模型众所周知的缺点：容易重复生成内容。在给出的示例中，GPT3 Davinci 生成了 14 个完全相同的小节，而人类鼓手则会在保持动机的同时“悄然加入”细微变化。值得注意的是，在流行乐、舞曲或摇滚乐等某些音乐流派中，重复鼓律动是可以接受的；但在大多数其他音乐流派中，这种做法并不受欢迎，因为人们通常认为它们听起来很机械。对于大语言模型生成的鼓律动而言，重复显然是应当避免的特征。

目前，我们采用无模型方法，大语言模型仅使用一些鼓律动数据进行训练，不包含任何额外信息或先验。为强化前文所述的优势并弥补其不足，我们建议未来的工作采用模块化方法，而不是以端到端方式处理任务。例如，可以从不同的分布中生成节奏型和加花，或使用不同的模型生成；还可以通过为每个小节过量生成内容并进行投票或选择，来明确抑制重复。此外，也可以在训练数据中注入额外标签，以此控制鼓律动中的变化度。

即兴创作还是背诵?

迄今为止，我们已暗示大语言模型在鼓乐创作方面具有创造力。这里，我们再提出一个问题：大语言模型真的在创作一些微调过程中从未见过的鼓律动，还是只是在复述它们已经见过的内容?

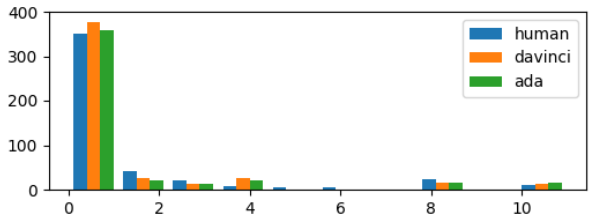

图 7：训练集中出现的每种生成小节的出现次数（纵轴）及其出现频率（横轴）。

我们通过计算测试集中每个生成小节在训练集中出现的频率来回答这个问题。如图 7 所示，只有少数生成小节与微调过程中见过的小节重复，这表明大语言模型能够创作新颖且未曾见过的鼓律动。

相关工作

自动音乐生成 历史悠久，近期研究主要关注人工智能（Kaliakatsos-Papakostas、Floros 和 Vrahatis，2020），尤其是深度神经网络的应用。这些研究不仅着眼于借助人工智能协助或取代人类作曲家和编曲家创作各种音乐，也旨在探究最先进的数据驱动模型的艺术创造力。当前大多数自动音乐生成研究都采用已证明在计算机视觉或自然语言处理领域有效的模型架构，例如长短期记忆网络（Lyu 等，2015）、Transformer（Huang 等，2018；Zeng 等，2021；Yu 等人（2022）），或用于学习嵌入空间的定制架构（Liang 等，2020）。然而，人工智能音乐生成的大部分研究都采用监督式方法，这极大地限制了其在较鲜为人知的音乐流派或特定乐器（例如鼓组生成）方面的应用。虽然目前尚无像 GPT-3 这样、与自然语言处理领域的对应模型同样通用的预训练音乐生成模型，但我们认为，探索语言到音乐的迁移是必要的，而这一领域仍未得到充分研究。

自动鼓生成现有研究数量不多。与一般的音乐生成类似，鼓乐生成任务可以在多种设定下进行。在最简单的设定中，只生成一个小节的鼓型（也称为“节拍”），并让它在整首歌曲中重复（Vogl and Knees 2017; Bruford, McDonald, and Sandler 2020; Tikhonov. and Yamshchikov. 2021）；而我们关注的是更复杂的非重复性鼓乐创作设定。另一些研究只考虑整体节奏（Lattner and Grachten 2019），而没有像我们这样强调不同鼓的配器。在更实际的设定中，研究者会根据贝斯线等音乐信号，生成较长的鼓乐创作序列（Makris et al. 2017, 2019）。这类研究与我们的工作最为相似，而我们的工作迄今只涉及鼓独奏表演。不过，上述所有研究使用的鼓乐创作数据规模和变化度都有限，所采用的模型（如长短期记忆网络）在人工智能领域也相对过时，尽管如此，未来的研究若能直接进行比较，将会有所裨益。另一类相关性较低的研究关注鼓演奏中的微时值和人性化处理（Gillick et al. 2019; Burloiu and Unatc 2020; Burloiu 2020），致力于模仿人类富有表现力的不完美。

少量研究关注节奏游戏（Donahue、Lipton 和 McAuley，2017；Liang、Li 和 Ikeda，2019）。尽管节奏游戏与击鼓有某些相似之处，例如都着重于根据节奏编排音符，但二者的核心区别在于：节奏游戏的编舞设计以难度和可玩性为优化目标，而鼓乐作曲则以音乐性为优化目标。鉴于这种根本性的动机差异，我们认为这类研究与我们的工作大多无关。

大语言模型（LLM）是深度神经网络模型，例如 Transformer，它们在海量文本语料上进行预训练。例如，GPT-3 的预训练数据来自 Common Crawl 的汇编，其中包含万维网上的大部分文本、公开书籍、维基百科等。从 BERT（Devlin 等，2019）到 GPT-3（Brown 等，2020），大语言模型在自然语言处理的大多数任务和应用中占据主导地位。尽管人们对大语言模型的工作原理知之甚少，因此它们也因其黑箱特性而闻名，但普遍认为，大语言模型的能力源于预训练数据和模型本身的庞大规模；这使大语言模型能够通过迁移学习，有效适应资源匮乏的领域，方法是在少量数据上进行微调。有趣的是，近期一些研究发现，这些能力还包括将预训练语言知识迁移到国际象棋等非语言任务中（St¨ockl，2021）。尽管国际象棋棋步或音乐谱等非语言文本表示表面上与自然语言相似，但二者的结构截然不同，因此我们认为研究它们之间的迁移学习很有价值。

结论与未来工作

我们的初步发现表明，经过微调的预训练大语言模型（LLM），即使仅使用数百个符号音乐文件（如鼓律动），也能够学会生成具有一定复杂度的音乐。我们还提供了证据，表明这种能力既与模型规模有关，也得益于语言预训练。我们希望这一发现能够激励研究者不仅探索低资源音乐生成，也进一步发掘大语言模型的非凡潜力。我们还尝试开创鼓律动自动评估的研究，希望借此推动未来人工智能鼓乐生成领域的工作。

我们还简要讨论了未来的研究计划。尽管文献中关于鼓乐生成的研究寥寥无几，且难以复现，我们仍将努力比较一些现有的专用模型。虽然我们当前的鼓点滚动表示忽略了力度，但可以通过将标记符‘o’替换为力度值，轻松编码这类信息。不过，这样做的效果仍有待探索。我们的方法或许可以移植到钢琴等其他乐器上，这是我们计划探索的方向；而要实现多乐器条件生成则会涉及更多工作。

参考文献

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020.《语言模型是少样本学习者》。载于 Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H. 编，《神经信息处理系统进展》，第 33 卷，1877–1901。Curran Associates, Inc.

Bruford, F.; McDonald, S.; and Sandler, M. 2020. jaki：使用 LSTM 编码器-解码器和深度强化学习，生成用户可控的鼓模式。载于*2020*年人工智能音乐创作联合会议。Brunner, G.; Wang, Y.; Wattenhofer, R.; and Zhao, S. 2018. 使用 CycleGAN 进行符号音乐流派迁移。载于*2018*年 *IEEE* 第*30*届人工智能工具国际会议（*ICTAI*），786–793。IEEE。Burloiu, G. 2020. 表现力鼓机中的微时序交互式学习。载于*2020*年人工智能音乐创作联合会议。Burloiu, G.; and Unatc, C. 2020. 通过迁移学习和循环神经网络实现自适应鼓机微时序。载于国际音乐信息检索学会会议（*ISMIR*）晚期成果演示环节扩展摘要。Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT：用于语言理解的深度双向变换器预训练。载于*2019*年北美计算语言学协会年会论文集：人类语言技术，第*1*卷（长篇与短篇论文），4171–4186。明尼阿波利斯，明尼苏达州：计算语言学协会。Donahue, C.; Lipton, Z. C.; and McAuley, J. 2017. 舞蹈舞蹈卷积。载于国际机器学习会议，1039–1048。PMLR。Gillick, J.; Roberts, A.; Engel, J.; Eck, D.; and Bamman, D. 2019. 通过逆序列变换学习节奏型。载于国际机器学习会议，2269–2279。PMLR。Huang, C.-Z. A.; Vaswani, A.; Uszkoreit, J.; Shazeer, N.; Simon, I.; Hawthorne, C.; Dai, A. M.; Hoffman, M. D.; Dinculescu, M.; and Eck, D. 2018. 音乐变换器。*arXiv* 预印本 *arXiv:1809.04281*。Kaliakatsos-Papakostas, M.; Floros, A.; and Vrahatis, M. N. 2020. 音乐生成的人工智能方法：综述与未来展望。自然启发计算与群体智能，217–245。Kuribayashi, T.; Oseki, Y.; Ito, T.; Yoshida, R.; Asahara, M.; and Inui, K. 2021. 困惑度较低并不总是更像人类。载于第*59*届计算语言学协会年会暨第*11*届自然语言处理联合国际会议论文集（第*1*卷：长篇论文），5203–5217。线上出版：计算语言学协会。Lattner, S.; and Grachten, M. 2019. 利用学得的节奏互动模式对鼓轨生成进行高层控制。载于*2019*年 *IEEE* 音频与声学信号处理应用研讨会（*WASPAA*），35–39。IEEE。Liang, H.; Lei, W.; Chan, P. Y.; Yang, Z.; Sun, M.; and Chua, T.-S. 2020. Pirhdy：学习用于符号音乐的音高、节奏和力度感知嵌入。载于第*28*届 *ACM* 国际多媒体会议论文集，574–582。Liang, Y.; Li, W.; and Ikeda, K. 2019. 使用深度学习方法进行节奏游戏的程序化内容生成。载于娱乐计算与严肃游戏联合国际会议，134–145。施普林格。Lyu, Q.; Wu, Z.; Zhu, J.; and Meng, H. 2015. 使用 LSTM-RTRBM 对高维序列建模：复调音乐生成的应用。载于第二十四届国际人工智能联合会议。Makris, D.; Kaliakatsos-Papakostas, M.; Karydis, I.; and Kermanidis, K. L. 2017. 将 LSTM 与前馈神经网络结合用于条件节奏创作。载于神经网络工程应用国际会议，570–582。施普林格。Makris, D.; Kaliakatsos-Papakostas, M.; Karydis, I.; and Kermanidis, K. L. 2019. 用于生成鼓节奏的条件神经序列学习器。神经计算与应用，31(6)：1793–1804。McMullen, E.; and Saffran, J. R. 2004. 音乐与语言：发展比较。音乐感知：跨学科期刊，21(3)：289–311。Roads, C. 1985. 音乐与人工智能研究。*ACM* 计算调查（*CSUR*），17(2)：163–190。St¨ockl, A. 2021. 观察语言模型学习国际象棋。载于自然语言处理最新进展国际会议论文集（*RANLP 2021*），1369–1379。线上举办：INCOMA Ltd.。Tikhonov., A.; and Yamshchikov., I. 2021. 人工神经网络随节拍即兴演奏。载于第六届复杂性、未来信息系统与风险国际会议论文集（*COMPLEXIS*），37–44。INSTICC，SciTePress。ISBN 978-989-758-505-0。Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. 注意力就是一切。载于 Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., 编，神经信息处理系统进展，第30卷。Curran Associates, Inc.。Vogl, R.; and Knees, P. 2017. 用于电子舞曲制作与演奏的智能鼓机。载于*NIME*，251–256。Yu, B.; Lu, P.; Wang, R.; Hu, W.; Tan, X.; Ye, W.; Zhang, S.; Qin, T.; and Liu, T.-Y. 2022. Museformer：采用细粒度与粗粒度注意力的音乐生成 Transformer。Zeng, M.; Tan, X.; Wang, R.; Ju, Z.; Qin, T.; and Liu, T.-Y. 2021. Musicbert：通过大规模预训练实现符号音乐理解。*arXiv* 预印本 *arXiv:2106.05630*。