# Simple and Controllable Music Generation

**Jade Copet**[♠◇] **Felix Kreuk**[♠◇] **Itai Gat** **Tal Remez** **David Kant**
**Gabriel Synnaeve** [◇] **Yossi Adi**[*◇] **Alexandre Défossez** [◇]
♠: equal contributions, ◇: core team
Meta AI
{jadecopet, felixkreuk, adiyoss}@meta.com

## Abstract

*警告：该PDF由GPT-Academic开源项目调用大语言模型+Latex翻译插件一键生成，版权归原文作者所有。翻译内容可靠性无保障，请仔细鉴别并以原文为准。
项目Github地址: https://github.com/binary-husky/gpt_academic/。
项目在线体验地址: https://chatpaper.org。
当前大语言模型: gpt-3.5-turbo，当前语言模型温度设定: 1。为了防止大语言模型的意外谬误产生扩散影响，禁止移除或修改此警告。

我们致力于条件音乐生成的任务。我们引入了 MusicGen，一个单一的语言模型 (Language Model, LM)，它能够处理多个流的离散压缩音乐表示，也就是说，令牌 (tokens)。与之前的研究不同，MusicGen由一个单级 Transformer LM 和高效的令牌交错模式组成，这消除了级联多个模型的需要，例如，分层或上采样。根据这种方法，我们展示了 MusicGen如何能够在条件文本描述或旋律特征的情况下生成高质量的样本，无论是单声道还是立体声，从而实现对生成输出的更好控制。通过大量的实证评估，包括自动和人为研究，我们展示了所提出的方法在一个标准的文本到音乐基准测试上优于评估基准。通过消融研究，我们阐明了构成 MusicGen的每个组成部分的重要性。音乐样本、代码和模型可在 github.com/facebookresearch/audiocraft 上获得。

## 1 Introduction

文本音乐生成是根据文本描述生成音乐作品的任务，例如“90年代摇滚歌曲带有吉他的断奏”。音乐生成是一项具有挑战性的任务，因为它需要对长范围序列进行建模。与语音不同，音乐需要使用全频谱 [Müller, 2015]。这意味着以更高的采样率采样信号，即音乐录音的标准采样率为44.1 kHz或48 kHz，而语音为16 kHz。此外，音乐包含来自不同乐器的和声和旋律，形成了复杂的结构。人类听众对不和谐的敏感度很高 [Fedorenko et al., 2012, Norman-Haignere et al., 2019]，因此在生成音乐时没有太多的余地可以犯旋律错误。最后，能够在各种方法中控制生成过程，例如音调、乐器、旋律、风格等，对于音乐创作者至关重要。

[*] Yossi Adi is Affiliated with both The Hebrew University of Jerusalem & MetaAI.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

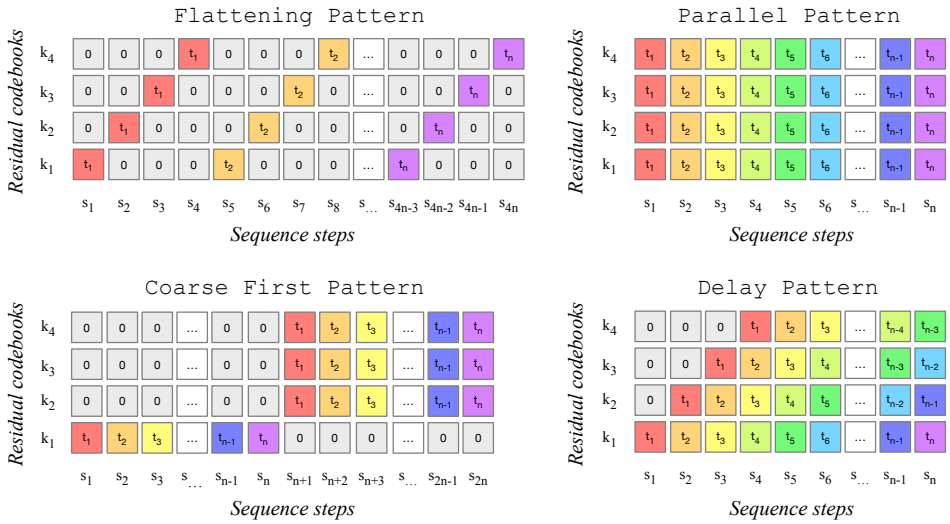



图 1: 代码本交错模式在第 2.2节中介绍。每个时间步$t_1, t_2, \ldots, t_n$由4个量化值组成（对应$k_1, \ldots, k_4$）。在进行自回归建模时，我们可以以各种方式对它们进行展开或交错，从而得到一个新的序列，其中有4个并行流和步骤$s_1, s_2, \ldots, s_m$。序列步长$S$的总数取决于模式和原始步长$T$。0是一个特殊的令牌，表示模式中的空位置。

最近在自监督音频表示学习 [Balestriero et al., 2023]、序列建模 [Touvron et al., 2023]和音频合成 [Tan et al., 2021]方面取得了进展，为开发此类模型提供了条件。为了使音频建模更具可操作性，最近的研究提出将音频信号表示为多个表示相同信号的离散令牌流 [Défossez et al., 2022]。这既可以实现高质量的音频生成，又可以实现有效的音频建模。然而，这样做的代价是同时建模几个并行相关的流。

Kharitonov et al. [2022], Kreuk et al. [2022]提出了一种并行建模多流语音令牌的延迟方法，即在不同流之间引入偏移。 Agostinelli et al. [2023]提出了使用离散令牌的多个序列以不同粒度表示音乐片段，并使用一系列自回归模型对其建模的方法。与此相似，Donahue et al. [2023]也采用了类似的方法，但用于歌唱伴奏生成任务。最近，Wang et al. [2023]提出了这个问题的两阶段解决方案：(i)仅建模第一个令牌流；(ii)然后，采用后网络以非自回归方式同时建模其余流。

在这项工作中，我们介绍了一个名为MusicGen的简单且可控的音乐生成模型，能够根据文本描述生成高质量的音乐。我们提出了一个模型多个并行声学令牌流的通用框架，这是对之前研究的一种泛化（见 Figure 1）。我们展示了如何通过这个框架实现对立体音频的扩展生成，而不增加额外的计算成本。为了改善生成样本的可控性，我们额外引入了无监督旋律调节，使模型能够生成与给定的和谐和旋律结构相匹配的音乐。我们对MusicGen进行了广泛的评估，并且通过主观评分来显示所提出的方法明显优于评估基准，MusicGen的得分为84.8，而最佳基准为80.5。我们还进行了消融研究，以阐明每个组成部分对整体模型性能的重要性。最后，人类评估表明，MusicGen生成的样本在旋律上与给定的和谐结构更加吻合，同时也遵循文本描述。

**我们的贡献：** (i) 我们引入了一个简单高效的模型，以32 kHz生成高质量音乐。我们证明了MusicGen可以通过高效的码本交错策略实现一阶段的一致音乐生成。（ii）我们提出了一个单一模型来执行文本和旋律条件的生成，并证明生成的音频与提供的旋律相一致，并忠实于文本条件信息。（iii）我们对我们方法的关键设计选择进行了广泛的客观和主观评估。

## 2 Method

MusicGen方法是基于自回归变压器解码器（[Vaswani et al., 2017]），其以文本或旋律表示为条件。该（语言）模型适用于一个来自EnCodec [Défossez et al., 2022]音频标记器的离散化单位，该标记器能够从低帧率的离散表示中提供高保真度的重构。压缩模型，例如[Défossez et al., 2022, Zeghidour et al., 2021]采用了残差向量量化（RVQ）方法，从而产生了多个并行流。在此设置下，每个流由来自不同的学习码书的离散令牌组成。之前的研究提出了几种处理这个问题的建模策略 [Kharitonov et al., 2022, Agostinelli et al., 2023, Wang et al., 2023]。在本工作中，我们引入了一种新的建模框架，可以推广到各种码书交错模式，并探索了几种变体。通过这些模式，我们可以利用量化音频令牌的内部结构。最后，MusicGen支持基于文本或旋律的条件生成。

### 2.1 Audio tokenization

我们使用EnCodec [Défossez et al., 2022]，这是一个使用残余向量量化（RVQ） [Zeghidour et al., 2021]对潜在空间进行量化的卷积自编码器，并采用对抗重建损失。给定一个参考音频随机变量$X \in \mathbb{R}^{d \cdot f_s}$，其中$d$表示音频时长，$f_s$表示采样率，EnCodec 将其编码成一个连续的张量，其帧率为$f_r \ll f_s$。然后，将这个表示量化为$Q \in \{1, \dots, M\}^{d \cdot f_r \times K}$，其中$K$是RVQ中使用的码书数量，$M$是码书的大小。注意，在量化之后，我们得到了$K$个平行的离散符号序列，每个序列的长度为$T = d \cdot f_r$，代表音频样本。在RVQ中，每个量化器编码前一个量化器留下的量化误差，因此不同码书的量化值一般来说是不独立的，而第一个码书是最重要的一个。

### 2.2 Codebook interleaving patterns (see Figure 1)

**精确展平自回归分解。** 自回归模型需要一个离散随机序列 $U \in \{1, \dots, N\}^S$，其中 $S$ 为序列长度。按照惯例，我们设定 $U_0 = 0$，作为一个确定性特殊标记，表示序列的起始。然后，我们可以对分布进行建模。

$$\forall t > 0, p_t\left(U_{t-1}, \dots, U_0\right) \triangleq \mathbb{P}\left[U_t | U_{t-1}, \dots, U_0\right]. \tag{1}$$

让我们通过使用自回归密度 $p$ 来构建第二个随机变量序列 $\tilde{U}$，例如，我们递归地定义 $\tilde{U}_0 = 0$，对于所有的 $t > 0$，

$$\forall t > 0, \mathbb{P}\left[\tilde{U}_t | \tilde{U}_{t-1} \dots, \tilde{U}_0\right] = p_t\left(\tilde{U}_{t-1}, \dots, \tilde{U}_0\right). \tag{2}$$

接下来，我们立即可以得出结论，$U$ 和 $\tilde{U}$ 服从相同的分布。这意味着，如果我们能够用深度学习模型拟合一个完美的估计 $\hat{p}$ 来对 $p$ 进行拟合，那么我们就可以完全拟合出 $U$ 的分布。

如前所述，我们从 EnCodec 模型得到的表示 $Q$ 的主要问题是，每个时间步骤都有 $K$ 个码书。一个解决方法是将 $Q$ 展平，即取 $S = d \cdot f_r \cdot K$，例如首先预测第一个时间步骤的第一个码书，然后是第一个时间步骤的第二个码书，依此类推。然后，使用方程 (1) 和方程 (2)，我们可以在理论上拟合出 $Q$ 的精确模型。然而，缺点是增加了复杂度，其中一部分增益来自最低采样率 $f_r$ 的丢失。

存在多种可能的展平方式，并且不需要通过单个模型来估计所有的 $\hat{p}_t$ 函数。例如，MusicLM [Agostinelli et al., 2023] 使用两个模型，一个模型对前 $K/2$ 个展平的码书建模，另一个模型对其余的 $K/2$ 个展平的码书建模，条件是根据第一个模型的决策。在这种情况下，自回归步骤的数量仍然是 $df_r \cdot K$。

**不精确的自回归分解。** 另一种可能性是考虑一种自回归分解方法，其中一些码书是在并行预测的。例如，让我们定义另一个序列，$V_0 = 0$，对于所有的 $t \in \{1, \ldots, T\}$，$k \in \{1, \ldots, K\}$，$V_{t,k} = Q_{t,k}$。当省略码书索引 $k$，例如 $V_t$，我们指的是时间 $t$ 的所有码书的串联。

$$p_{t,k}\left(V_{t-1}, \ldots, V_0\right) \triangleq \mathbb{P}\left[V_{t,k} | V_{t-1}, \cdot, \ldots, V_0\right]. \tag{3}$$

让我们再次递归地定义 $\tilde{V}_0 = 0$，并且对于所有的 $t > 0$，

$$\forall t > 0, \forall k, \mathbb{P}\left[\tilde{V}_{t,k}\right] = p_{t,k}\left(\tilde{V}_{t-1}, \ldots, \tilde{V}_0\right). \tag{4}$$

与公式 (2) 中不同的是，在一般情况下，$\tilde{V}$ 不再服从与 $V$ 相同的分布，即使我们假设我们可以获取精确的分布 $p_{t,k}$。实际上，只有当对于所有 $t$，在已知 $V_{t-1}, \ldots, V_0$ 的条件下，$(V_{t,k})_k$ 是独立的时，我们才有一个合适的生成模型。随着 $t$ 的增加，误差会累积，两个分布之间的差异会增大。这样的分解是不精确的，但可以保留原始帧率，可以大大加快训练和推断的速度，特别是对于长序列。

**任意码本交错模式。** 为了试验各种这样的分解方式，并准确测量使用不精确分解的影响，我们引入了*码本交错模式*。让我们考虑 $\Omega = \{(t, k) : \{1, \ldots, d \cdot f_r\}, k \in \{1, \ldots, K\}\}$ 是所有时间步和码本索引对的集合。码本模式是一个序列 $P = (P_0, P_1, P_2, \ldots, P_S)$，其中 $P_0 = \emptyset$，对于所有 $0 < s \leq S$，$P_s \subset \Omega$，且 $P$ 是 $\Omega$ 的一个划分。我们通过在所有 $P_0, P_1, \ldots, P_{s-1}$ 的位置的条件下，预测并行地在 $P_s$ 的所有位置上的 $Q$。实际上，我们限制模式，使得每个码本索引在任何一个 $P_s$ 中最多只出现一次。

现在我们可以很容易地定义许多分解方式，例如所谓的“并行”模式是指

$$P_s = \{(s, k) : k \in \{1, \ldots, K\}\}. \tag{5}$$

还可以像 Kharitonov et al. [2022] 中那样，在码书之间引入“延迟”。例如，

$$P_s = \{(s - k + 1, k) : k \in \{1, \ldots, K\}, s - k \geq 0\}. \tag{6}$$

通过实证评估，我们展示了各种码本模式的优点和缺点，揭示了对并行码本序列进行精确建模的重要性。

### 2.3 Model conditioning

**文本条件。**给定与输入音频$X$匹配的文本描述，我们计算一个条件张量$C \in \mathbb{R}^{T_C \times D}$，其中$D$是自回归模型中使用的内部维度。通常，有三种主要的方法用于表示用于有条件音频生成的文本。Kreuk et al. [2022]提出了使用预训练文本编码器，具体来说是T5 [Raffel et al., 2020]。Chung et al. [2022]表明，使用基于指令的语言模型提供了更好的性能。最后，Agostinelli et al. [2023], Liu et al. [2023], Huang et al. [2023a], Sheffer and Adi [2023]声称，联合文本-音频表示，如CLAP [Wu* et al., 2023]，提供了更高质量的生成结果。我们分别尝试了上述所有方法：T5编码器、FLAN-T5和CLAP。

**旋律条件。**虽然文本是当今有条件生成模型的主要方法，但对于音乐来说，一种更自然的方法是基于另一段音频的旋律结构或者吹口哨或者哼唱进行条件建模。这种方法还允许对模型的输出进行迭代性的改进。为了支持这一点，我们尝试通过基于输入的色度图和文本描述的联合条件来控制旋律结构。在初步实验中，我们观察到仅仅在原始色度图上进行条件建模往往导致重构原始样本，从而导致过拟合。为了减少这种情况，我们在每个时间步选择主导的时频 bin，引入了一个信息瓶颈。虽然Agostinelli et al. [2023]展示了类似的功能，但那些作者使用了受限制的专有数据，这种数据的收集是费时费力的。在本文中，我们采用了无监督的方法，消除了对受监督数据的需求。

### 2.4 Model architecture

**编码本投影和位置嵌入。** 给定一个编码本模式，在每个模式步骤 $P_s$ 中只有一部分编码本是存在的。我们从 $Q$ 中取出与 $P_s$ 中的索引相对应的值。如 Section 2.2 中所述，每个编码本最多在 $P_s$ 中出现一次，或者根本不出现。如果它存在，则我们使用一个有 $N$ 个条目和维度 $D$ 的学习嵌入表来表示与 $Q$ 相关联的值。否则，我们使用一个特殊的符号表示其缺失。在进行这种转换后，我们对每个编码本的贡献求和。由于 $P_0 = \emptyset$，第一个输入总是所有特殊符号的总和。最后，我们添加一个正弦嵌入来编码当前步骤 $s$ [Vaswani et al., 2017]。

**Transformer解码器。** 输入被送入一个具有 $L$ 层和维度 $D$ 的Transformer。每一层都由一个因果自注意块组成。然后，我们使用一个交叉注意块，其中输入为条件信号 $C$。当使用旋律条件时，我们将条件张量 $C$ 作为前缀提供给Transformer输入。该层以一个全连接块结束，该块由一个线性层（从 $D$ 到 $4 \cdot D$ 通道），一个ReLU激活函数和一个线性层（返回到 $D$ 通道）组成。注意力和全连接块都包含一个残差跳跃连接。在与残差跳跃连接相加之前，对每个块进行层归一化 [Ba et al., 2016]（“预归一化”）。

**Logits预测。** Transformer解码器在模式步骤 $P_s$ 的输出被转换为对于由 $P_{s+1}$ 给出的索引处的 $Q$ 的logits预测值。每个编码本在 $P_{s+1}$ 中最多出现一次。如果编码本存在，则通过将具有 $D$ 通道到 $N$ 的线性层应用于获得logits预测。

## 3 Experimental setup

### 3.1 Models and hyperparameters

**音频分词模型。**我们使用非因果五层的32 kHz 单声道音频编码模型，步长为640，每秒50帧，在模型的五个层中初始隐藏大小为64，每个层都会翻倍。嵌入使用四个量化器进行量化，每个量化器的码本大小为2048。我们遵循Défossez et al. [2022]的方法，在音频序列中以随机方式进行裁剪来训练模型。

**Transformer模型。**我们训练了不同规模的自回归Transformer模型：300M、1.5B、3.3B 参数模型。我们使用内存高效的Flash attention [Dao et al., 2022]，即xFormers软件包 [Lefaudeux et al., 2022]进行加速和减少长序列的内存使用。我们在Section 4中研究了模型规模对性能的影响。我们使用300M参数模型进行所有的实验。我们随机从完整音轨中采样30秒的音频片段进行训练。我们使用AdamW优化器 [Loshchilov and Hutter, 2017]进行1M步的训练，批量大小为192个样本，$\beta_1 = 0.9$，$\beta_2 = 0.95$，解耦权重衰减为0.1，梯度裁剪为1.0。对于300M模型，我们进一步使用基于D-Adaptation的自动步长调整 [Defazio and Mishchenko, 2023]，以提高模型的收敛性，但对于更大的模型则没有带来收益。我们使用余弦学习率调度，热身阶段为4000步。此外，我们使用指数移动平均法进行模型优化，衰减率为0.99。我们使用32、64和96个GPU分别训练300M，1.5B和3.3B参数模型，并使用混合精度训练。具体而言，我们使用float16精度，而bfloat16在我们的设置中会导致不稳定。最后，在采样时，我们采用top-k采样方法 [Fan et al., 2018]，保留前250个标记，采样温度为1.0。

**文本预处理。**Kreuk et al. [2022]提出了一种文本归一化方案，其中省略了停用词，剩余的文本进行词形还原。我们将该方法称为文本归一化。在考虑音乐数据集时，通常会提供额外的标注标签，如音乐关键字、节奏、乐器类型等。我们还尝试将这些标注与文本描述进行拼接。我们将这种方法称为条件合并。最后，我们尝试使用词丢失作为另一种文本增强策略。对于最终的模型，我们使用概率为0.25的条件合并。在合并后，我们以概率0.5进行文本描述丢弃。我们对得到的文本使用概率为0.3的词丢失。

表 1: 文本到音乐生成。 我们将MUSICGEN与多个基准模型进行客观和主观度量的比较。我们报告平均分数和CI95分数。Mousai模型在相同的数据集上重新训练， 而对于MusicLM，我们使用公共API进行人类研究。我们报告了Noise2Music和MusicLM在MusicCaps上的原始FAD。“MUSICGEN w. random melody”指的是使用色度谱和文本进行训练的MUSICGEN。在评估时，我们从一个保留集合中随机采样色度谱。

| | MUSICCAPS Test Set | | | | |
|---|---|---|---|---|---|
| MODEL | $\text{FAD}_{vgg}\downarrow$ | KL $\downarrow$ | $\text{CLAP}_{scr}\uparrow$ | OVL. $\uparrow$ | REL. $\uparrow$ |
| Riffusion | 14.8 | 2.06 | 0.19 | 79.31±1.37 | 74.20±2.17 |
| Mousai | 7.5 | 1.59 | 0.23 | 76.11±1.56 | 77.35±1.72 |
| MusicLM | 4.0 | - | - | 80.51±1.07 | 82.35±1.36 |
| Noise2Music | **2.1** | - | - | - | - |
| MUSICGEN w.o melody (300M) | 3.1 | 1.28 | 0.31 | 78.43±1.30 | 81.11±1.31 |
| MUSICGEN w.o melody (1.5B) | 3.4 | 1.23 | **0.32** | 80.74±1.17 | **83.70**±1.21 |
| MUSICGEN w.o melody (3.3B) | 3.8 | **1.22** | 0.31 | **84.81**±0.95 | 82.47±1.25 |
| MUSICGEN w. random melody (1.5B) | 5.0 | 1.31 | 0.28 | 81.30±1.29 | 81.98±1.79 |

关于不同的文本预处理策略的详细比较可以在Appendix A.2中找到。

**码本模式和条件导入。** 我们使用Section 2.2中的“延迟”交错模式， 将30秒的音频转换为1500个自回归步骤。对于文本条件导入，我们使用T5 [Raffel et al., 2020]文本编码器，可选地加入在Section 2.3中介绍的旋律条件。我们还尝试了FLAN-T5 [Chung et al., 2022]和CLAP [Wu* et al., 2023]，并在Appendix A.2中比较了每种文本编码器在MUSICGEN中的性能。对于旋律条件，我们使用窗口大小为$2^{14}$和跳跃大小为$2^{12}$进行计算色谱图。使用较大的窗口可以防止模型恢复细节。我们通过在每个时间步骤上取最大值来量化色谱图。我们遵循Kreuk et al. [2022]的方法，在从模型的对数中进行采样时，实现了无分类器的指导。具体而言，在训练期间以概率0.2丢弃条件，在推断期间使用指导缩放因子为3.0。

### 3.2 Datasets

**训练数据集。** 我们使用20K小时的授权音乐来训练MUSICGEN。具体而言，我们依赖于内部数据集中的10K首高质量音乐曲目，并依赖于ShutterStock和Pond5音乐数据集[2]，分别包含25K和365K个仅含乐器的音乐曲目。所有数据集都包含以32 kHz采样的完整音乐，并附带文本描述、流派、BPM和标签等元数据信息。除非另有说明，我们将音频混合为单声道。

**评估数据集。**对于主要结果和与之前的工作进行比较，我们在MusicCaps基准测试上评估了所提出的方法[Agostinelli et al., 2023]。MusicCaps由5.5K个由专业音乐家准备的样本（时长为10秒）组成，其中包括了平衡的1K个样本跨越多种流派。我们在不平衡的集合上报告客观评估指标， 而在平衡的流派集合中，我们从中抽样进行定性评估。对于旋律评估和消融研究，我们使用来自域内保留评估集的528首音乐曲目的样本，与训练集中的艺术家没有重复。

[2] www.shutterstock.com/music和www.pond5.com

### 3.3 Evaluation

**基线方法。** 我们将MusicGen与文本生成音乐的两种基线进行比较：Riffusion [Forsgren and Martiros]和Mousai [Schneider et al., 2023]。我们使用开源的Riffusion模型来进行推断[3]。对于Mousai，我们使用我们的数据集训练了一个模型，以便进行公平的比较，并使用了作者提供的开源实现[4]。另外，如果可能的话，我们还与MusicLM [Agostinelli et al., 2023]和Noise2Music [Huang et al., 2023b]进行比较。

**评估指标。** 我们使用客观和主观指标来评估所提出的方法。对于客观方法，我们使用了三个指标：Fréchet音频距离（FAD），Kullback-Leibler散度（KL）和CLAP得分（CLAP）。我们使用Tensorflow中VGGish模型的官方实现报告了FAD [Kilgour et al., 2018]。FAD得分低表示生成的音频是可信的。按照 Kreuk et al. [2022]的做法，我们使用了一个在AudioSet [Koutini et al., 2021]上进行分类训练的最先进音频分类器来计算原始音乐和生成音乐之间标签概率的KL散度。当KL值较低时，生成的音乐与参考音乐之间应具有相似的概念。最后，使用官方预训练的CLAP模型 [5]计算了轨道描述与生成音频之间的CLAP得分，以量化音频文本对齐情况。

对于人类研究，我们遵循了 Kreuk et al. [2022]的相同设置。我们要求人类评估员评估音频样本的两个方面：（i）整体质量（Ovl）和（ii）与文本输入的相关性（Rel）。对于整体质量测试，评估员被要求在1到100的范围内评价提供的样本的感知质量。对于文本相关性测试，评估员被要求在1到100的范围内评价音频与文本之间的匹配程度。我们使用亚马逊机械土耳其平台招募评估员。我们评估了随机抽样的文件，每个样本至少由5名评估员进行评估。我们使用CrowdMOS软件包[6]来过滤嘈杂的注释和异常值。我们删除没有完整听取录音的注释者，对参考录音评分低于85的注释者，以及CrowdMOS [Ribeiro et al., 2011]的其余建议。为了公平起见，所有样本均在−14dB LUFS [ITU-R, 2017]进行了归一化处理。

## 4 Results

我们首先介绍了提出的方法在文本至音乐生成任务上的结果，并与该领域的先前工作进行了比较。接下来，我们评估了提出方法在以旋律特征为条件生成音乐方面的能力。我们进一步展示了如何简单地扩展我们的代码本模式来生成立体声音频。最后，我们进行了一项消融研究。音乐示例、代码和模型可在github.com/facebookresearch/audiocraft上获得。

### 4.1 Comparison with the baselines

表 1呈现了所提出方法与Mousai、Riffusion、MusicLM和Noise2Music的比较结果。由于Noise2Music没有官方实现和预训练模型，我们仅报告了MusicCaps上的FAD（见Noise2Music的文稿）。类似地，MusicLM的实现也没有公开。我们在主观测试中使用了MusicLM的公开演示[7]，同时报告了作者提供的FAD。尽管原始MusicLM模型是基于包含人声的数据进行训练的，但API后端使用的是仅含乐器的模型。对于人类研究，我们只限制在从MusicCaps中选择的40个仅有乐器的样本上进行。为了防止MusicGen在使用色度图进行训练时出现泄漏问题，在测试时我们从一个保留的集合中随机采样色度图。

结果表明，MusicGen在音频质量和对提供的文本描述的遵守度方面优于评估的基准，得到了人类听众的认可。在MusicCaps上，Noise2Music在FAD方面表现最佳，其次是在文本调整

[3]使用来自github.com/riffusion/riffusion-app的riffusion-model-v1（于2023年5月10日）
[4]来自github.com/archinetai/audio-diffusion-pytorch（2023年3月）
[5]https://github.com/LAION-AI/CLAP
[6]http://www.crowdmos.org/download/
[7]https://blog.google/technology/ai/musiclm-google-ai-test-kitchen/

表 2: 我们报告了参考曲调和生成曲调之间的余弦相似度（SIM.），以及包括与曲调的对齐程度（MEL.）在内的主观评估指标。所有结果均基于MUSICGEN 1.5B模型。

| | | In Domain Test Set | | | |
|---|---|---|---|---|---|
| TRAIN CONDITION | TEST CONDITION | SIM. ↑ | MEL. ↑ | OVL. ↑ | REL. ↑ |
| Text | Text | 0.10 | 64.44±0.83 | 82.18±1.21 | 81.54±1.22 |
| Text+Chroma | Text | 0.10 | 61.89±0.96 | 81.65±1.13 | **82.50**±0.98 |
| Text+Chroma | Text+Chroma | **0.66** | **72.87**±0.93 | **83.94**±1.99 | 80.28±1.06 |

条件下训练的MUSICGEN。有趣的是，添加旋律调整会降低客观指标，但对人类评分影响不大，同时仍优于评估的基准。

我们注意到，对于评分最低的模型，FAD与整体主观评分存在相关性，但对评分最高的模型却不是如此。我们注意到，MusicCaps [Agostinelli et al., 2023]中有大量样本表明录音存在“噪音”。当生成音频的质量达到一定阈值时，由于这些噪音样本，改善音频质量可能会降低MusicCaps上的FAD。

### 4.2 Melody evaluation

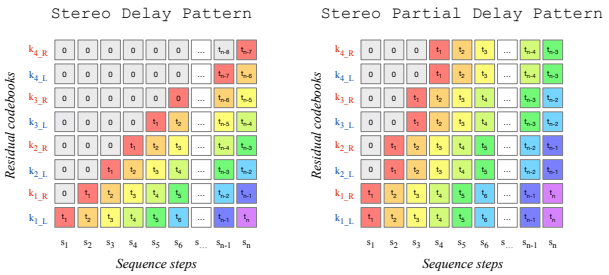
我们使用客观和主观指标对MUSICGEN进行评估，同时以文本和旋律表示为条件，评估保留的评估集。对于客观评估，我们引入了一种新的度量方式：色度余弦相似度，它衡量了来自参考和生成样本的量化色度对应时间步的帧之间的平均余弦相似度。我们使用从一个保留集中随机抽取的1000个文件进行评估。为了更好地评估条件旋律与生成音乐之间的关系，我们进行了另一项人类研究。为此，我们向人类评定者呈现一首参考音乐作品，然后是一组生成作品。对于每个生成的样本，听众被要求在1到100的范围内评估生成作品的旋律与参考旋律的匹配程度。我们从保留集中随机选择了40个10秒钟的样本。结果见表Table 2。结果表明，通过色度图条件训练的MUSICGEN成功生成了符合给定旋律的音乐，因此可以更好地控制生成的输出。有趣的是，在推理时去掉色度，OVL.和REL.几乎不变，说明MUSICGEN具有鲁棒性。

### 4.3 Fine-tuning for stereophonic generation

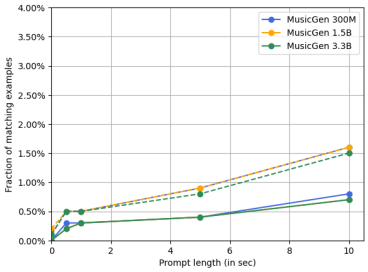
我们尝试扩展声音生成至立体声数据。我们使用相同的EnCodec 分词器，独立地应用于左右声道，每帧提供$2 \cdot K = 8$个码本。我们从预先训练的单声道MUSICGEN模型开始，利用立体声音频在相同数据集上进行了20万批次的微调。我们重用了“延迟”模式，有两种可能的变体：(i) “立体声延迟”在同一码本级别上引入了左右声道之间的延迟；(ii) “立体声部分延迟”对于给定级别，对左右声道的码本应用相同的延迟，如图 2a所示。请注意，使用这种简单的策略，我们可以以**无额外计算成本**生成立体声音频。我们在表 3中提供了对这些模型的主观评价。我们可以看到，当将立体声输出混合为单声道时，其感知质量几乎与单声道模型相当。立体声音频总体上比单声道对应物评分更高，“立体声部分延迟”在整体质量和文本相关性方面比“立体声延迟”模式有了小幅提升。

### 4.4 Ablation

本节提供了对不同码本模式的割蚀研究，以及模型规模和记忆研究的结果。另外，我们还在 Appendix A.2中提供了不同文本增强策略、文本编码器和音频分词模型的结果。所有割蚀实验均使用从留置评估集中随机抽样的30秒，共计1K个样本进行。




(a) 使用两个可能的交错方式来可视化双目模型的码本模式。对于给定的码本索引，“立体声延迟”模式为左声道和右声道使用不同的延迟，而“立体声部分延迟”模式则同时预测两个声道。




(b) 在从训练集中提取的不同持续时间的片段提示下，考虑到确切匹配（实线）和80% 部分匹配（虚线），对于5秒音频生成的第一个码本标记的记忆结果。

图 2: 立体声码本（左）和记忆结果（右）

表 3: 立体声文本到音乐生成。 EnCodec将左右声道分别处理，得到8个码书而不是4个。我们尝试了两种码书模式，如图 2a所示。我们还测量了将立体声模型混音成单声道后的一个模型。我们使用了一个仅基于文本条件的1.5B的MusicGen模型。

| | | MusicCaps Test Set | |
|---|---|---|---|
| Cb. Pattern | Stereo? | Ovl. ↑ | Rel. ↑ |
| *Mono Delay* | ✗ | 84.95±1.60 | **80.61**±1.22] |
| *Stereo Partial Delay* | ✗* | 84.49±1.80 | 79.39±1.16 |
| *Stereo Partial Delay* | ✓ | **86.73**±1.06 | **80.41**±1.15 |
| *Stereo Delay* | ✓ | 85.51±1.21 | 78.32 ±1.21 |

*: downmixed to mono after generation.

表 4: 码本模式。我们对音频序列的30秒进行了不同的码本交织模式比较。“压平”模式取得了最佳得分。“延迟”和“部分压平”模式获得了相似的得分，而“并行”模式得分较差。

| | | In Domain Test Set | | | | |
|---|---|---|---|---|---|---|
| Configuration | Nb. steps | $FAD_{vgg}$ ↓ | KL ↓ | $CLAP_{scr}$ ↑ | Ovl. ↑ | Rel. ↑ |
| Delay | 1500 | 0.96 | 0.52 | 0.35 | **79.69**±1.46 | 79.67±1.41 |
| Partial delay | 1500 | 1.51 | 0.54 | 0.32 | 79.13±1.56 | 79.67±1.46 |
| Parallel | 1500 | 2.58 | 0.62 | 0.27 | 72.21±2.49 | 80.30±1.43 |
| Partial flattening | 3000 | 1.32 | 0.54 | 0.34 | 78.56±1.86 | 79.18±1.49 |
| Coarse first | 3000 | 1.98 | 0.56 | 0.30 | 74.42±2.28 | 76.55±1.67 |
| Flattening | 6000 | **0.86** | **0.51** | **0.37** | **79.71**±1.58 | **82.03**±1.1 |

**码本交织模式的效果。**我们使用来自 Section 2.2 的框架评估了各种码本模式，其中$K = 4$，由音频标记模型给出。 Table 1 报告了使用“延迟”模式的结果。“部分延迟”是指将码本 2、3 和 4 都延迟相同的数量。“并行”模式在同一时间步骤中预测所有码本。“粗糙优先”模式首先为所有步骤预测码本 1，然后并行预测码本 2、3 和 4。因此，与其他模式相比，这种模式的步骤数是其他模式的两倍。“部分展平”类似，但是在为所有步骤采样码本 1 时，它将其与并行采样的码本 2、3 和 4 交错。最后，“展平”模式是将所有码本展平，类似于

表 5: 模型规模。我们比较了三个规模的模型，并在内部测试集上进行评估，以限制我们在MusicCaps中观察到的域外预测问题的影响。在客观指标方面，我们观察到指标的持续改进，尽管主观质量在15亿个参数时停止改进。然而，33亿个参数的模型似乎更适合文本提示。

| | | | | In Domain Test Set | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Dim. | Heads | Depth | # Param. | PPL↓ | $\text{FAD}_{\text{vgg}}$ ↓ | KL ↓ | $\text{CLAP}_{\text{scr}}$ ↑ | OVL. ↑ | REL. ↑ |
| 1024 | 16 | 24 | 300M | 56.1 | 0.96 | 0.52 | 0.35 | 78.3±1.4 | 82.5 ±1.6 |
| 1536 | 24 | 48 | 1.5B | 48.4 | 0.86 | **0.50** | 0.35 | **81.9**±1.4 | 82.9±1.5 |
| 2048 | 32 | 48 | 3.3B | **46.1** | **0.82** | **0.50** | **0.36** | 79.2±1.3 | **83.5**±1.3 |

MusicLM [Agostinelli et al., 2023]。我们在Table 4中报告了客观和主观评估。展平可以提高生成质量，但计算成本很高，而使用简单的延迟方法也可以达到类似的效果，但成本只是展平方法的一小部分。

**模型规模的影响。**我们在Table 5中报告了不同模型规模（300M、1.5B 和 3.3B参数模型）的结果。如预期的，调整模型规模会得到更好的分数，但代价是更长的训练和推断时间。关于主观评估，1.5B 的整体质量最佳，但较大的模型能更好地理解文本提示。

**记忆实验。**根据[Agostinelli et al., 2023]，我们分析了 MUSICGEN的记忆能力。我们仅考虑来自 MUSICGEN的第一个代码流，因为它包含较粗粒度的信息。我们从训练集中随机选择 $N = 20,000$ 个例子，对于每个例子，我们使用原始音频和调制信息对应的 EnCodec 码本来提供模型的提示。我们使用贪婪解码生成了 250 个音频标记（5 秒音频）的延续。我们报告确切匹配作为生成的音频标记和源音频标记完全相同的例子的比例。此外，我们报告了部分匹配，作为训练样本中生成序列和源序列至少有 80% 的音频标记匹配的比例。当改变音频提示的长度时，我们在图 2b 中呈现了不同模型规模的记忆结果。

## 5 Related work

**音频表示。**近年来，一种突出的方法是将音乐信号表示为压缩表示，离散或连续，并在其上应用生成模型。Lakhotia等人（2021）提出使用k-means量化语音表示来构建语音语言模型。最近，Defossez等人（2022）和Zeghidour等人（2021）提出直接在原始波形上应用VQ-VAE，并使用残差向量量化。随后，几项研究使用这种表示进行文本到音频生成。接下来，我们讨论音频生成方面的最新研究。

**音乐生成。** 音乐生成在各种设置下已经研究了很长时间。 Dong等人 （2018） 提出了基于GAN的符号音乐生成方法。Bassan等人（2022）提出了一种用于符号音乐的无监督分割方法，可以用于后续生成。Ycart等人（2017）提出了使用递归神经网络进行多声部音乐建模。Ji等人（2020）对音乐生成的深度学习方法进行了全面的调查。

Dhariwal等人（2020）提出使用分层VQ-VAE将音乐样本表示为多个离散表示流。接下来，两个稀疏变压器应用于序列上以生成音乐。Gan等人（2020）提出了为给定视频生成音乐，并预测其midi音符。最近，Agostinelli等人（2023）提出使用多个“语义令牌”和“声学令牌”来表示音乐。然后，他们应用一系列的变压器解码器，以基于文本-音乐联合表示进行条件建模。Donahue等人（2023）采用了类似的建模方法，但用于合唱伴奏生成任务。

一个替代方法是使用扩散模型。 Schneider等人 （2023）、 Huang等人 （2023）、 Maina等人 （2023） 和Forsgren等人 （扩散） 提出了一种潜在扩散模型用于文本到音乐的任务。Schneider等人 （2023） 提出了使用扩散模型进行音频编码器-解码器和潜在生成的方法。

Huang等人（2023）提出了一系列扩散模型，用于生成音频并逐渐增加其采样率。Forsgren等人（扩散）提出使用频谱图微调Stable Diffusion（Rombach等人，2022），以生成五秒的片段，然后使用图像到图像映射和潜在插值生成长序列。

**音频生成。**已经提出了几项用于文本到音频（环境声音）生成的研究。Yang等人（2022）提出使用VQ-VAE表示音频谱图，然后在生成部分使用以文本CLIP嵌入为条件的离散扩散模型。Kreuk等人（2022）提出应用变压器语言模型于通过使用EnCodec （Defossez等人，2022）直接量化时域信号得到的离散音频表示。Sheffer等人（2023）采用了类似于Kreuk等人（2022）的方法进行图像到音频生成。Huang等人（2023）、Liu等人（2023）提出使用潜在扩散模型用于文本到音频的任务，同时将其扩展到各种其他任务，如修复、图像到音频等。

## 6 Discussion

我们引入了一种先进的单阶段可控音乐生成模型MusicGen，该模型可以在文本和旋律的条件下进行生成。我们证明了简单的码本交错策略可以用来实现高质量的生成，即使在立体声的情况下，同时相对于平铺方法，减少了自回归时间步的数量。我们对模型大小、条件方法和文本预处理技术的影响进行了全面的研究。我们还引入了一种基于色谱图的简单条件方法来控制生成音频的旋律。

**局限性** 我们的简单生成方法不能对生成结果与条件的精细控制，主要依赖于成本函数的指导。此外，尽管进行文本条件增强相对较直接，但对音频进行条件增强需要进一步研究数据增强、指导类型和数量。

**更广泛的影响** 大规模生成模型存在伦理挑战。首先，我们确保所训练的所有数据都经过了与版权持有者的合法协议，特别是与ShutterStock的协议。第二个方面是我们所使用的数据集可能缺乏多样性，其中包含更多西方风格的音乐。然而，我们相信在这项工作中进行的简化，例如使用单阶段语言模型和较少的自回归步骤，可以帮助将应用扩展到新的数据集中。生成模型可能对艺术家构成不公平竞争，这是一个公开的问题。开放的研究可以确保所有参与者都能平等地访问这些模型。通过引入我们所提出的更高级的控制，例如旋律条件方法，我们希望这些模型对音乐业余爱好者和专业人士都能有用。

**伦理声明** 本文在哈马斯发动的一起恐怖袭击之后完成，这次袭击让以色列国家深感震惊。于2023年10月7日，数千名哈马斯恐怖分子渗透以色列边境，对22个以色列村庄发动了激烈的袭击，残忍地夺走了一千多条无辜生命，并绑架了两百多名平民。

在我们哀悼和悲痛我们的朋友和家人的同时，我们呼吁学术界团结一致，谴责哈马斯犯下的这些不可言喻的暴行，并为被绑架人员的迅速和安全归还倡导，我们共同追求和平。

纪念那些被哈马斯行径摧毁的无数生命。

## Acknowledgements.

作者们要感谢Mary Williamson、Rashel Moritz和Joelle Pineau对本项目的支持，感谢Justin Luk、Prash Jain、Sidd Srinivasan、Rod Duenes和Philip Woods提供的数据集，以及感谢xformers团队的Daniel Haziza、Francisco Massa和Michael Ramamonjisoa进行的技术讨论。

## 参考文献


Meinard Müller. *Fundamentals of music processing: Audio, analysis, algorithms, applications*, volume 5. Springer, 2015.

Evelina Fedorenko, Josh H McDermott, Sam Norman-Haignere, and Nancy Kanwisher. Sensitivity to musical structure in the human brain. *Journal of neurophysiology*, 108(12):3289–3300, 2012.

Sam V Norman-Haignere, Nancy Kanwisher, Josh H McDermott, and Bevil R Conway. Divergence in the functional organization of human and macaque auditory cortex revealed by fmri responses to harmonic tones. *Nature neuroscience*, 22(7):1057–1060, 2019.

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*, 2021.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Eugene Kharitonov, Ann Lee, Adam Polyak, Yossi Adi, Jade Copet, Kushal Lakhotia, Tu Anh Nguyen, Morgane Riviere, Abdelrahman Mohamed, Emmanuel Dupoux, et al. Text-free prosody-aware generative spoken language modeling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8666–8681, 2022.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. *arXiv preprint arXiv:2209.15352*, 2022.

Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

Chris Donahue, Antoine Caillon, Adam Roberts, Ethan Manilow, Philippe Esling, Andrea Agostinelli, Mauro Verzetti, Ian Simon, Olivier Pietquin, Neil Zeghidour, et al. Singsong: Generating musical accompaniments from singing. *arXiv preprint arXiv:2301.12662*, 2023.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551, 2020.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*, 2022.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023.

Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. *arXiv preprint arXiv:2301.12661*, 2023a.

Roy Sheffer and Yossi Adi. I hear your true colors: Image guided audio generation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

Yusong Wu*, Ke Chen*, Tianyu Zhang*, Yuchen Hui*, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, 2022.

Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. xformers: A modular and hackable transformer modelling library. https://github.com/facebookresearch/xformers, 2022.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Aaron Defazio and Konstantin Mishchenko. Learning-rate-free learning by d-adaptation. *arXiv preprint arXiv:2301.07733*, 2023.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*, 2018.

S Forsgren and H Martiros. Riffusion-stable diffusion for real-time music generation. 2022. *URL https://riffusion. com/about*.

Flavio Schneider, Zhijing Jin, and Bernhard Schölkopf. Mo\^ usai: Text-to-music generation with long-context latent diffusion. *arXiv preprint arXiv:2301.11757*, 2023.

Qingqing Huang, Daniel S Park, Tao Wang, Timo I Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jiahui Yu, Christian Frank, et al. Noise2music: Text-conditioned music generation with diffusion models. *arXiv preprint arXiv:2302.03917*, 2023b.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fr\'echet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

Khaled Koutini, Jan Schlüter, Hamid Eghbal-zadeh, and Gerhard Widmer. Efficient training of audio transformers with patchout. *arXiv preprint arXiv:2110.05069*, 2021.

Flávio Ribeiro, Dinei Florêncio, Cha Zhang, and Michael Seltzer. Crowdmos: An approach for crowdsourcing mean opinion score studies. In *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2416–2419. IEEE, 2011.

ITU-R. Algorithms to measure audio programme loudness and true-peak audio level, 2017.

Kushal Lakhotia, Eugene Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Abdelrahman Mohamed, et al. On generative spoken language modeling from raw audio. *Transactions of the Association for Computational Linguistics*, 9:1336–1354, 2021.

Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Shahaf Bassan, Yossi Adi, and Jeffrey S Rosenschein. Unsupervised symbolic music segmentation using ensemble temporal prediction errors. *arXiv preprint arXiv:2207.00760*, 2022.

Adrien Ycart, Emmanouil Benetos, et al. A study on lstm networks for polyphonic music sequence modelling. ISMIR, 2017.

Shulei Ji, Jing Luo, and Xinyu Yang. A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions. *arXiv preprint arXiv:2011.06801*, 2020.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

Chuang Gan, Deng Huang, Peihao Chen, Joshua B Tenenbaum, and Antonio Torralba. Foley music: Learning to generate music from videos. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 758–775. Springer, 2020.

Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel PW Ellis. Mulan: A joint embedding of music audio and natural language. *arXiv preprint arXiv:2208.12415*, 2022.

Kinyugo Maina. Msanii: High fidelity music synthesis on a shoestring budget. *arXiv preprint arXiv:2301.06468*, 2023.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022.

Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *arXiv preprint arXiv:2207.09983*, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Alexandre Défossez, Nicolas Usunier, Léon Bottou, and Francis Bach. Music source separation in the waveform domain. *arXiv preprint arXiv:1911.13254*, 2019.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *arXiv preprint arXiv:2306.06546*, 2023.

将「figure」计数器与「section」相关联。将「table」计数器与「section」相关联。

# A Appendix

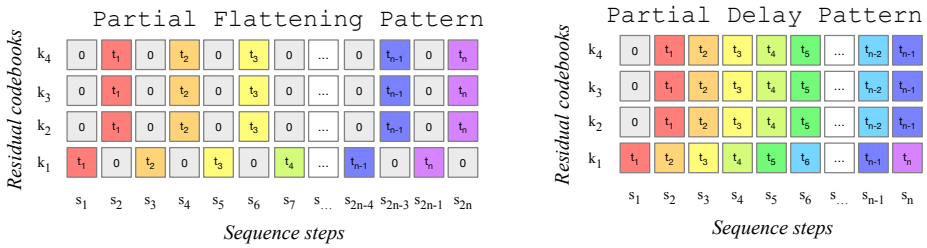



图 A.1: 对一个具有4个并行流量化值序列（对应着$k_1$，...，$k_4$）和$N$个时间步长（$t_1$，...，$t_n$）的序列应用部分平铺和部分延迟码本模式进行可视化。“部分平铺”将第一个码本分开，并将其与码本2、3和4的并行采样交错，导致交错序列步长$M$是原始步长$N$的两倍。“部分延迟”模式是指将码本2、3和4同时延迟相同的量，我们这里使用1个单位的延迟。交错序列的总步数为$N$（为简单起见，不计入最后一步）。

## A.1 Experimental details

**码本交织模式。** Figure A.1提供了在Section 4中进行剔除（abbration）所引入的额外码本模式的可视化描述，包括“局部平坦化”和“局部延迟”模式。这些模式背后的思想是，RVQ的第一个码本是最重要的，通过并行预测其他码本可以限制引入的平坦化或延迟，同时保持良好的建模性能。

**旋律条件化。** 在这项工作中，我们提供了一种基于音色图（chromagram）表示的无监督旋律条件化方法。如Figure A.2所示，基于音色图的条件化在生成新的音乐样本时成功地保持了旋律结构。在初步的实验中，我们注意到音色图往往被低频乐器所主导，主要是鼓和低音。为了缓解这个问题，我们使用了Demucs [Défossez et al., 2019] 来将参考音轨分解为四个组成部分：鼓、贝斯、人声和其他。然后，我们省略了鼓和贝斯，以恢复残差波形的旋律结构。最后，我们提取量化的音色图，以创建之后馈送给模型的条件化输入。

**流派分布。** 我们在图 A.3 中提供了数据集中前50个音乐流派的直方图。我们注意到，Dance/EDM流派占主导地位，我们的经验也表明这是MusicGen最好生成的流派之一。虽然我们尝试了一些重新采样方案来提高其他流派的重要性，但我们观察到，过度采样其他代表较少的流派往往会导致整体模型更差。

## A.2 Additional experimental results

我们在MusicGen的核心组件上进行了进一步的消融研究，即用于文本调节描述的文本编码器（详见Section 2.3），文本增强策略（详见Section 3.1）和所使用的音频标记模型。我们在MusicCaps数据集上报告了结果，以更好地了解不同方法的域外泛化能力。最后，我们还分享了优化方法的其他实验结果。

表 A.1: 文本编码器结果。我们报告了T5、Flan-T5和CLAP作为文本编码器的结果。我们观察到T5和Flan-T5在所有客观指标上的结果相似。请注意，T5是用于主要MUSICGEN模型的文本编码器。 CLAP编码器在所有指标上的表现一直较差，除了CLAP分数。所有比较都是基于仅使用文本调节的300M MUSICGEN模型进行的。

| | MUSICCAPS Test Set | | | | |
|---|---|---|---|---|---|
| MODEL | FAD$_{vgg}$ ↓ | KL ↓ | CLAP$_{scr}$ ↑ | OVL. ↑ | REL. ↑ |
| T5 | **3.12** | **1.29** | 0.31 | 85.04±1.23 | **87.33±1.9** |
| Flan-T5 | 3.36 | 1.30 | 0.32 | **85.54±1.01** | 85.00±1.63 |
| CLAP | 4.16 | 1.36 | 0.35 | 82.13±1.29 | 83.56±1.54 |
| CLAP (no normalization) | 4.14 | 1.38 | 0.35 | 84.87±1.25 | 85.06±1.72 |
| CLAP (no quantization) | 5.07 | 1.37 | **0.37** | 84.13±1.02 | 84.67±1.42 |

**文本编码器的影响。** 我们研究了文本编码器的影响，比较了三种不同的编码器：T5 [Raffel et al., 2020] [8]，Flan-T5 [Chung et al., 2022][9] 和CLAP [Wu* et al., 2023] [10]，其中CLAP带有量化瓶颈。对于基于CLAP的编码器，类似于Agostinelli et al. [2023]，我们在训练过程中依赖音乐嵌入，并在推理时使用文本嵌入，然后在提取的嵌入之上训练一个RVQ层。具体来说，我们首先对嵌入进行归一化处理，然后使用RVQ进行量化，其中包含12个量化器，每个量化器的码书大小为1024。通过对CLAP嵌入进行量化，可得到一种均匀的表示，离散标记进一步减小了训练时使用的音频编码与测试时使用的文本编码之间的差距。我们在 Table A.1 中报告了不同文本编码器的结果。在客观指标方面，T5和Flan-T5表现相似，总体质量稍微优于T5。然而，基于CLAP的模型在客观和主观指标方面表现较差，CLAP分数除外，该分数依赖于相同的音频和文本编码器。

**文本扩充的效果。** 我们研究了文本增强策略对所提出方法的影响。具体而言，我们研究了条件合并（即将额外的元数据连接到文本描述中）、文本归一化（text-norm.）和词丢弃的使用。我们在Table A.2中报告了不同增强策略的客观指标。我们观察到，在使用条件合并结合额外元数据时，FAD和KL有所提升。然而，无论是文本归一化还是词丢弃都没有在客观指标方面改善结果。

**音频分词器的效果。** 我们实验中替换了EnCodec为Descript音频编解码器(DAC) [Kumar et al., 2023][11]，这是一种类似的音频压缩模型，它使用了不同的训练集，通过多波段鉴别器增强了类似的对抗性损失，并在较低维度空间进行量化以提高码本使用效率。DAC以44.1 kHz的采样率进行音频压缩，使用9个码本和86 Hz的帧率。我们使用DAC和EnCodec作为音频分词器，在我们数据集的无人声版本上训练了一个小型（3亿参数）和中等型（15亿参数）的MUSICGEN模型。表 A.3中提供的结果显示，我们的领域内测试集上的FAD和KL都变差了。在MusicCaps上，使用DAC可以改善FAD，但KL变差，以及主观评价。在本研究中使用的EnCodec模型专门设计为以较低的帧率（50 Hz）运行，而DAC的帧率为86 Hz，因此为了生成相同的音频，其推理运行时间降低了40%。最后需要注意的是，DAC是在与EnCodec不同的数据集上训练的，还需要进一步的实验来了解这些压缩模型对自回归语言模型适应度的影响。

[8] https://huggingface.co/t5-base

[9] https://huggingface.co/google/flan-t5-base

[10] https://huggingface.co/lukewys/laion_clap/blob/main/music_audioset_epoch_15_esc_90.14.pt

[11] 使用公开实现github.com/descriptinc/descript-audio-codec.

表 A.2: 文本增强策略结果。我们报告了仅使用原始文本描述（无增强）以及不同的文本增强策略的客观指标：使用条件合并来将元数据补充到文本描述中，使用文本标准化（文本标准化）并对结果文本应用词汇抛弃。我们使用了训练了500K步的300M MusicGen模型。条件合并改善了仅对原始文本描述进行训练的结果。其他增强在所有指标上表现最差。我们在主要模型中使用条件合并与词汇抛弃相结合的方式，展现了最佳的文本相关性。

| | MusicCaps Test Set | | | | |
|---|---|---|---|---|---|
| Configuration | $\text{FAD}_{vgg}\downarrow$ | KL ↓ | $\text{CLAP}_{scr}$ | Ovl. ↑ | Rel. ↑ |
| No augmentation | 3.68 | 1.28 | **0.31** | **83.40**±1.44 | 81.16±1.29 |
| Condition Merging (CM) | **3.28** | **1.26** | **0.31** | 82.60±1.41 | 84.45 ±1.16 |
| CM + Text-norm. (TN) | 3.78 | 1.30 | 0.29 | 80.57±2.14 | 82.40±1.09 |
| CM+ Word dropout (WD) | 3.31 | 1.31 | 0.30 | 82.52±1.55 | **85.27**±0.97 |
| CM + TN + WD | 3.41 | 1.39 | 0.30 | 81.18±1.91 | 84.32 ±1.59 |

表 A.3: 我们用 DAC [Kumar et al., 2023] 替换 EnCodec，并使用他们的实现进行测试。 DAC 是一个 44.1 kHz 的模型，具有 9 个码书和 86 Hz 的帧率。这些模型是在我们数据集的无人声版本上进行训练的，所以客观度量指标与其他表中报告的指标可能不匹配。我们同时在我们的域测试集和 MusicCaps [Agostinelli et al., 2023] 上报告客观度量指标，仅在 MusicCaps 上报告主观度量指标。

| | In Domain Test Set | | MusicCaps Test Set | | |
|---|---|---|---|---|---|
| Model | $\text{FAD}_{\text{vgg}}\downarrow$ | KL ↓ | $\text{FAD}_{\text{vgg}}\downarrow$ | KL ↓ | Ovl. ↑ |
| MusicGen + DAC small | 3.45 | 0.58 | 4.46 | 1.35 | 83.32±0.95 |
| MusicGen + DAC medium | 2.42 | 0.57 | **4.32** | 1.30 | 84.46±0.97 |
| MusicGen + EnCodec small | 0.67 | 0.54 | 5.26 | 1.27 | 84.69±0.90 |
| MusicGen + EnCodec medium | **0.49** | **0.52** | 5.05 | **1.23** | **86.09**±0.88 |

**D适应性的效果。**

D适应性是一种新颖的自动化方式，用于在整个训练过程中动态选择Adam优化器的整体学习率，即其$\alpha$参数，由Defazio and Mishchenko [2023]引入。我们观察到在3亿参数模型中改进了收敛性，但对于更大的模型，例如15亿和33亿，我们观察到自动规则导致了训练集和验证集的性能下降。需要进一步调查以更好地理解D适应性的效果，以及它是否适用于最大的模型。图 A.4中可以观察到两种方法在训练集和验证集上的收敛情况。

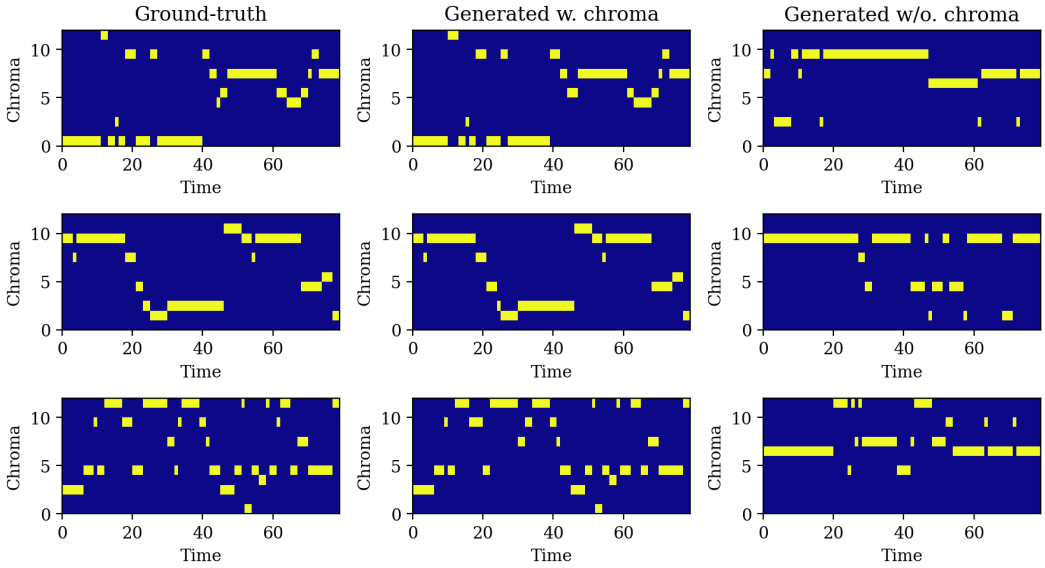



图 A.2: 在参考旋律上，通过时间可视化量化过的色度图的二进制。由色度和文本进行条件生成的音乐位于中间，只有文本进行条件生成的音乐位于右侧。每一行都使用不同的色度条件生成，但所有行都共享相同的文本条件：“90年代摇滚歌曲，电吉他和重型鼓”。我们观察到使用色度条件生成的音乐样本对输入旋律非常忠实，同时也在已有文本的指导下呈现出新颖的风格。

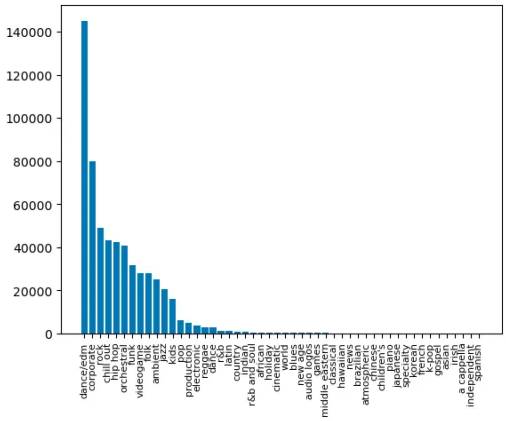


图 A.3: 训练数据集中排名前50的音乐流派的直方图。

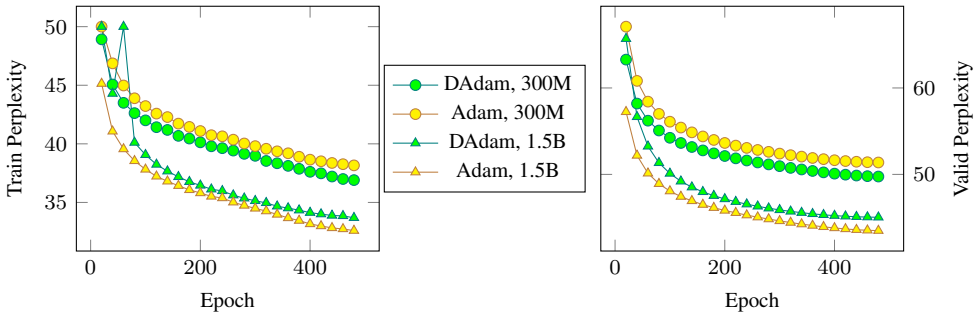



图 A.4: Adam和Adam with D-Adaptation的比较（Defazio等人，2023）。尽管D-Adaptation对于300M参数模型提供了一致的收益，但我们观察到1.5B参数模型在训练（左侧）和验证（右侧）集上的收敛性较差。