arXiv:2301.06468v1 [cs.SD] 16 Jan 2023

Msanii：低预算高保真音乐合成

进行中工作（WiP）

Kinyugo Maina独立研究者
kinyugomaina@gmail.com

2023年1月18日

摘要

本文介绍了 Msanii，这是一种新颖的基于扩散的模型，可高效合成长上下文、高保真音乐。我们的模型结合了梅尔频谱图的表现力、扩散模型的生成能力以及神经声码器的声码能力。我们通过合成数十秒（*190* 秒）的立体声高采样率音乐（*44.1kHz*），且不使用拼接合成、级联架构或压缩技术，展示了 Msanii 的有效性。据我们所知，这是首个成功使用基于扩散的模型，在高采样率下合成如此长音乐片段的研究。我们的演示可在此处查看，我们的代码也可在此处查看。

1 引言

音乐是一种能够激发情感、连接不同文化背景人群的通用语言，也是社会不可或缺的一部分。数十年来，研究人员一直在探索计算机能否捕捉音乐创作背后的创作过程，以及这对音乐和人工智能可能产生的影响。

近年来，生成式建模领域取得了显著发展，涌现出多种技术，包括生成对抗网络（GANs）Goodfellow 等人（2020）、变分自编码器（VAEs）Kingma 和 Welling（2013）、归一化流 Rezende 和 Mohamed（2015）、自回归模型 Dhariwal 等人（2020）以及扩散模型 Sohl-Dickstein 等人（2015）；Ho 等人（2020）；Kingma 等人（2021），推动了各个领域的进步。这些技术在人类水平的任务表现上已取得显著成果，例如图像生成 Rombach 等人（2022）；Karras 等人（2020）；Dhariwal 和 Nichol（2021）；Saharia 等人（2022）；Ramesh 等人（2022）、语音生成 Kong 等人（2020）；Shen 等人（2017）以及文本生成 Brown 等人（2020）；Scao 等人（2022），同时还在推动了音乐生成Dhariwal 等人 (2020)；Rouard 和 Hadjeres (2021)；Engel 等人(2019)；Marafi- otietal. (2019)以及其他领域。

然而，由于音频信号具有高维度，模型难以学习音乐的长程结构，因此，高效的高保真音乐合成仍然是机器学习中的一项挑战Dhari- wal 等人 (2020）；Dieleman 等人 (2018）。为解决这一问题，通常会学习音频信号的低维表示，从而降低计算复杂度，并使模型能够更好地捕捉与保真度相关的音乐显著特征Dhariwal 等人(2020）；Dieleman 等人(2018）。作为学习低维表示的替代方案，时频（TF）表示（如梅尔频谱图）为表示音频信号中的特征提供了一种强大且直观的方法。梅尔频谱图作为一种时频特征，已广泛应用于各种任务，包括自然语言处理 Shen 等人 (2017）、语音转换 Hwang 等人 (2020）以及歌声合成Liu 等人 (2022a）。它们也已成功应用于

音乐合成 Engel 等人 (2019）； Marafioti 等人(2019）； Pasini 和 Schlüter (2022）。梅尔频谱图对于音乐合成尤其具有吸引力，因为其分辨率较低，能够在降低计算复杂度的同时捕捉重要的音乐特征。

自回归模型和生成对抗网络是音乐合成的热门选择，但各自也面临着挑战。自回归模型广泛应用于原始波形域 Dhariwal 等人 (2020);Dieleman 等人 (2018), 推理速度往往较慢。生成对抗网络常用于基于时频表示的音乐合成 Engel 等人 (2019);Marafioti 等人 (2019); Pasini 和 Schlüter (2022)，但可能会遭遇训练不稳定的问题；此外，多网络的对抗训练会导致样本多样性较低，而且计算成本高昂。相比之下，基于扩散的模型相较于自回归模型推理速度更快，训练流程简单，并且近期在质量方面超越了生成对抗网络 Dhariwal 和 Nichol(2021)。因此，基于扩散的模型是音乐合成的一种极具吸引力的选择。

本文提出了一种利用梅尔频谱图进行音乐合成的新方法，该方法充分发挥了基于扩散的建模的优势。通过结合梅尔频谱图的表现力、我们新颖的 U-Net 架构以及扩散模型，我们能够以高采样率合成数分钟的高保真音乐。我们的方法代表了音乐合成领域的一项重大进展：它无需依赖拼接合成、级联架构或压缩技术，即可生成时长较长的高品质音乐样本。此外，我们还表明，我们的模型 Msanii 无需重新训练，即可用于解决其他音频任务，例如音频修补和风格迁移。

本文的主要贡献如下：

- •我们介绍了 Msanii，这是一种新颖的基于扩散的模型，可在梅尔频谱图域中合成长上下文、高保真音乐。据我们所知，这是首项成功在时频域中使用基于扩散的模型，以高采样率（44.1 kHz）合成数分钟音频的工作。
- • 我们通过合成数十秒（*190* 秒）高采样率（*44.1 kHz*）的立体声音乐，展示了 Msanii 的有效性。
- • 我们展示了 Msanii 可用于解决其他音频任务，例如插值、风格迁移，内绘和外绘，无需重新训练。

本文其余部分安排如下：第2 节回顾音乐合成和基于扩散的模型相关工作。第3 节介绍我们提出的方法，包括 Msanii 的架构和训练。第4 节介绍实验设置。第5 节展示实验结果。第6 节讨论未来工作。最后，第7 节总结我们的贡献。

免责声明：本文仍在持续完善中，尚未定稿。本文所呈现的结果可能会有所变动，不应视为最终结果。

2 背景

音频信号的高维性给原始波形域中的音乐合成带来了重大挑战。为了准确表示一个音频样本，需要将连续信号离散化为大量样本，这就要求使用较高的采样率（每秒数千个样本）。对于 CD 音质的音乐，采样率通常为 44.1 kHz。这意味着，一段时长为~3 分钟的音频样本大约包含~8 百万个样本。声道数量越多，复杂度也越高，因为样本总数 T会变为 $T = duration \times sample\ rate \times channels$。

合成长音频样本的计算需求，还会因需要捕捉丰富多样的音乐结构（如音色、和声和旋律），并确保曲式和织体方面的全局连贯性而进一步增加。为应对这些挑战，通常会使用音频信号的低维表示，在降低计算复杂度的同时捕捉重要的音乐特征。此外，还需要采用一种表现力强且高效的生成模型。

2.1 梅尔频谱图

为了解决任务的计算复杂度问题，我们提出使用一种维度较低但表现力丰富的音频表示：梅尔频谱图。梅尔频谱图是一种广泛应用于语音合成等任务的音频表示Shen 等人 (2017）、语音转换Hwang 等人(2020）和音乐合成 Vasquez 和 Lewis (2019）。它们源自短时傅里叶变换（STFT）的幅度频谱图，并以一种更符合人类听觉感知的方式编码频率。然而，原始

音频转换为梅尔频谱图并非完全可逆，因为幅度短时傅里叶变换频谱图中的相位信息有所损失。

为了从梅尔频谱图重建音频，研究人员开发了 MelGAN 等神经声码器Kumar 等人(2019)、ISTFTNetKaneko 等人 (2022)、MCNN Arık 等人 (2018) 和相位梯度 Di Giorgi 等人 (2022)，旨在从梅尔频谱图中同时近似幅度和相位。然而，要设计一种轻量级声码器，同时重建幅度和相位，可能颇具挑战。作为替代方案，我们提出仅重建幅度频谱图，并使用传统方法近似相位。以往的研究已展示了这种方法，例如对抗式音频合成 Marafioti 等人(2019) 和 MelNet Vasquez 和 Lewis (2019)。具体而言，我们采用 Griffin-Lim 算法 Griffin 和 Lim(1984)；Perraudin 等人 (2013)。

为了重建幅度频谱图，我们结合使用谱收敛损失和对数幅度损失。谱收敛损失定义如下：

$$\frac{|||STFT(s)| - |STFT(s)|||_F}{|||STFT(s)|||_F} \tag{1}$$

其中，$|.|$ 表示幅度，$|.|_F$ 是弗罗贝尼乌斯范数，s和 $\hat{s}$ 分别是真实值和预测的幅度频谱图。此损失侧重于频谱图中幅度较大的分量。

对数幅度损失定义如下：

$$\|log(|STFT(s)| + \epsilon) - log(|STFT(\hat{s})| + \epsilon)\|_1 \tag{2}$$

其中，ϵ是一个较小的常数值，用于避免对零取对数；$|.|_1$ 是 $L1$ 范数。此损失侧重于频谱图中幅度较小的分量。

2.2 扩散

受近期扩散模型在解决音频任务方面取得的成功启发 Sohl- Dickstein 等人(2015)；Ho 等人 (2020)；Kingma 等人(2021) 在音频任务中的应用 Rouard 和 Hadjeres (2021)；Kong 等人 (2020)，我们选择将其用于合成梅尔频谱图。扩散模型可以被视为一种马尔可夫层次变分自编码器 Luo (2022)。它们定义了一个马尔可夫链，通过一系列步骤逐渐向数据添加随机噪声，然后学习反向过程，以从噪声合成数据样本。

2.2.1 前向过程

给定一个真实的数据分布 $q(\boldsymbol{x}_0)$，我们从该分布中抽取一个样本$\boldsymbol{x}_0$，$\boldsymbol{x}_0 \sim q(\boldsymbol{x})$。然后，我们定义一个前向加噪过程 $q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$，按照预先设定的时间表 $\beta_t \in (0,\ 1)_{t=1}^T$逐渐向样本添加高斯噪声，其中 $\beta_1 < \beta_2 < \cdots < \beta_T$。具体来说，每个时间步的样本分布为：

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\mathbf{I}) \tag{3}$$

给定初始样本 $\boldsymbol{x}_0$，整个样本序列$\boldsymbol{x}_{1:T}$的分布为：

$$q(\boldsymbol{x}_{1:T}|\boldsymbol{x}_0) = \prod_{t=1}^{T} q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) \tag{4}$$

当 $T \to \infty$ 时，$\boldsymbol{x}_T$ 的分布会趋近于标准高斯分布。

要从前向分布中采样，我们可以在每个时间步 t 从均值为 $\mu_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}$、方差为 $\sigma^2 = \beta_t$ 的条件高斯分布中抽取一个样本 $\boldsymbol{x}_t$，如下所示：

$$\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon} \quad \text{where } \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{5}$$

扩散模型中的前向加噪过程具有这样的性质：利用重参数化技巧，我们可以在任意时间步采样 $\boldsymbol{x}_t$。这一性质由 Song 等人(2020)；Luo(2022)；Weng (2021) 描述。令 $\alpha_t = 1 - \beta_t$ 和 $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$。该过程可表示如下：

$$\begin{aligned} \boldsymbol{x}_t &= \sqrt{\alpha_t}\boldsymbol{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1} \\ &= \sqrt{\alpha_t\alpha_{t-1}}\boldsymbol{x}_{t-2} + \sqrt{1-\alpha_t\alpha_{t-1}}\bar{\boldsymbol{\epsilon}}_{t-2} \\ &= \dots \\ &= \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon} \end{aligned} \tag{6}$$

其中，$\boldsymbol{\epsilon}_{t-1},\ \boldsymbol{\epsilon}_{t-2},\ \cdots \sim \mathcal{N}(\mathbf{0},\ \mathbf{I})$。$\bar{\boldsymbol{\epsilon}}_{t-2}$是两个方差不同的高斯分布的组合，分别为 $\mathcal{N}(\mathbf{0},\ \sigma_1^2\mathbf{I})$ 和 $\mathcal{N}(\mathbf{0},\ \sigma_2^2\mathbf{I})$，满足：

$$\bar{\boldsymbol{\epsilon}}_{t-2} = \sqrt{(1-\alpha_t) + \alpha_t(1-\alpha_{t-1})} = \sqrt{1-\alpha_t\alpha_{t-1}} \tag{7}$$

2.2.2 反向过程

要从高斯噪声输入 $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0},\ \mathbf{I})$ 生成一个样本，我们通过从$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ 采样来逆转前向过程。然而，$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ 取决于整个数据集，因此我们学习一个近似模型 p_θ：

$$p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{x}_{t-1}; \mu_\theta(\boldsymbol{x}_t, t), \Sigma_\theta(\boldsymbol{x}_t, t)) \quad (8)$$

$$p_\theta(\boldsymbol{x}_{0:T}) = p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) \quad (9)$$

当以 $\boldsymbol{x}_0$ 为条件时，逆向条件概率变得易于处理：

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_{t-1}; \tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\Sigma}_t(\boldsymbol{x}_t, \boldsymbol{x}_0)) \quad (10)$$

与Ho 等人（2020）类似，我们将方差固定为：

$$\Sigma_\theta(\boldsymbol{x}_t, \boldsymbol{x}_0) = \sigma_t^2 \mathbf{I} = \tilde{\beta}_t \mathbf{I} \quad (11)$$

并且仅学习均值 μ_θ。由此得到：

$$q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_{t-1}; \tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0), \tilde{\beta}_t \mathbf{I}) \quad (12)$$

利用贝叶斯定理，我们可以推导出以下方程：

$$\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t}\boldsymbol{x}_t + \frac{\sqrt{\bar{\alpha}_{t-1}\beta_t}}{1-\bar{\alpha}_{t-1}}\boldsymbol{x}_0 \quad (13)$$

$$\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t \quad (14)$$

将 $\boldsymbol{x}_0 = \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t - \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t)$ 代入上述关于 $\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0)$ 的方程，可得：

$$\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_t) \quad (15)$$

该方程使我们能够在给定 $\boldsymbol{x}_t$ 和 $\boldsymbol{x}_0$的情况下，计算分布 $q(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t, \boldsymbol{x}_0)$的均值。该分布的方差~固定为$\beta_t\mathbf{I}$，其中 β_t 由方程 $\tilde{\beta}_t = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t} \cdot \beta_t$给出。随后，我们可以从该分布中采样，在给定 $\boldsymbol{x}_t$ 和$\boldsymbol{x}_0$的情况下得到 $\boldsymbol{x}_{t-1}$ 的一个样本。重复这一过程，即可在给定 $\boldsymbol{x}_T$的情况下生成$\boldsymbol{x}_{t-2}$，$\boldsymbol{x}_{t-3}$、……、$\boldsymbol{x}_0$ 的样本。

2.2.3 损失函数

由于训练期间可以获取 $\boldsymbol{x}_t$，我们可以利用它对高斯分布的均值项进行重参数化，并在给定$\boldsymbol{x}_t$ 和 t的情况下预测 $\boldsymbol{\epsilon}_t$。具体通过以下方程实现：

$$\tilde{\mu}_t(\boldsymbol{x}_t, \boldsymbol{x}_0) = \frac{1}{\sqrt{\bar{\alpha}_t}}(\boldsymbol{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)) \quad (16)$$

然后，我们使用 Ho 等人（2020）所述的简化损失项。由此得到以下方程：

$$\begin{aligned} L_t &= \mathbb{E}_{t\sim[1,T],\boldsymbol{x}_0,\boldsymbol{\epsilon}_t}[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\|_2^2] \\ &= \mathbb{E}_{t\sim[1,T],\boldsymbol{x}_0,\boldsymbol{\epsilon}_t}[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\alpha_t}\boldsymbol{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_t, t)\|_2^2] \end{aligned} \quad (17)$$

3 架构

受视觉 Transformer（VIT）中图块嵌入词元化方案成功的启发，Dosovitskiy 等人（2020），我们提出了一种基于梅尔频谱图的类似音频词元化方案。具体而言，我们将梅尔频谱图视为一系列词元，其中序列长度等于时间帧数，每个词元的维度等于梅尔频率的数量。这类似于沿频率维度提取图块，图块大小等于梅尔频率的数量。

与其他将梅尔频谱图视为图像的方法不同，我们的方法通过缩小上下文大小来提高效率。对于尺寸为$channels \times frequencies \times time\ frames$的梅尔频谱图，我们将上下文大小缩减至 $channels \times time\ frames$。此外，我们对各个声道分别进行处理，因此该方法适用于任意数量的声道。这些考量已融入我们的 U-Net（见第3.1节）和神经声码器（见第3.2节）的设计中。

3.1 U-Net

U-Net Ronneberger 等人（2015）凭借跳跃连接，能够准确建模细节并定位特征，因此成为图像分割任务的热门选择。这些神经网络也广泛应用于扩散建模Nichol 和 Dhariwal（2021）；Song 等人（2020）；Ho 等人（2020）。在这项工作中，我们提出了一种 U-Net 架构，将 U-Net 的优势与 Transformer 的优势相结合 Vaswani 等人（2017），后者以能够通过自注意力机制捕捉长程依赖而著称。由此得到的模型能够捕捉局部和全局上下文，同时保持高效（见图1）。下文将介绍 U-Net 的组成部分。

3.1.1 输入层和输出层

输入层（分词层）接收一个二维频谱图 $\mathbf{X} \in \mathbb{R}^{c\times f\times l}$，其中 c 为通道维度（例如单声道、立体声），f 为频率维度，l 为时间帧维度。它沿频率维度对其进行重塑和线性投影，以获得 $\mathbf{H} \in \mathbb{R}^{d\times c\times t}$，其中 $\mathbf{H}$ 为输出，d 为投影后的频率维度。

输出层（反分词层）接收输入$\mathbf{H} \in \mathbb{R}^{d\times c\times l}$，并执行与输入层相反的操作，将输入进行线性投影并重塑为$\mathbf{Y} \in \mathbb{R}^{c\times f\times l}$，其中$\mathbf{Y}$ 是输出（见图1）。

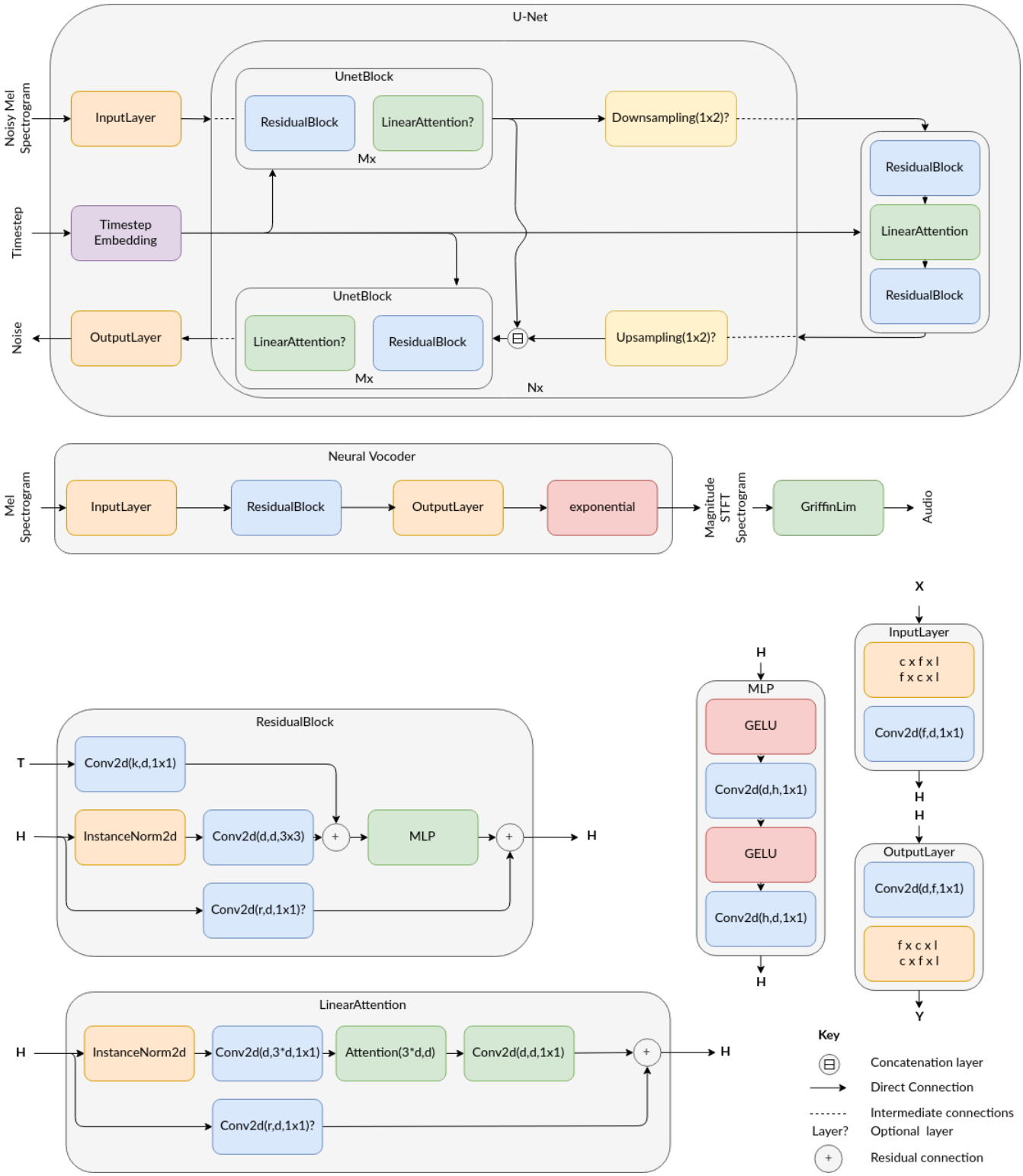

图 1：提出的 Msanii架构。

3.1.2 残差块

残差块的输入是一个二维潜在特征 $\mathbf{H} \in \mathbb{R}^{d \times c \times l}$ 和一个时间步嵌入特征 $\mathbf{T} \in \mathbb{R}^{k \times 1 \times 1}$，其中 d 是频率维度，c 是通道维度，l 是时间帧维度，而 k 是时间步。我们首先对输入特征应用预归一化层，然后执行 3×3 填充卷积，且不改变特征数量，以确保输入和输出特征具有相同的维度。虽然 U-Net 对所用归一化的具体类型并不敏感，但我们发现，特征归一化的效果优于权重归一化。这可能是因为特征归一化会使预激活呈高斯分布，从而使输出也呈高斯分布，这有利于对通常呈高斯分布的噪声进行建模。我们在实现中选择使用实例归一化 Ulyanov 等人(2016)，因为已有观察表明，在我们的神经声码器中，这种方法生成的音频金属伪影更少3.2。

对输入特征进行归一化和投影后，我们使用 1×1 卷积投影时间步嵌入特征，使其维度与潜在特征的维度相匹配。然后，我们将时间步嵌入特征与潜在特征相加，并将结果输入多层感知机（MLP）。我们的 MLP 基于 ConvNext Liu 等人（2022b），并做了一处小幅修改：在第一个卷积层之前应用 GELU 非线性函数。最后，我们对 MLP 的输出应用残差连接，并可选择将残差特征投影到与 MLP 输出相匹配的维度。

3.1.3 线性注意力

线性注意力模块的输入是一个二维潜在特征张量 $\mathbf{H} \in \mathbb{R}^{d \times c \times l}$。我们首先对输入张量进行实例归一化，然后执行注意力运算。尽管注意力机制的选择对 U-Net 的性能影响并不显著，但我们发现，线性注意力 Katharopoulos 等人（2020）由于具有线性计算复杂度，表现良好。

与一些 Transformer 不同Vaswani 等人（2017）；Doso- vitskiy 等人 （2020）；Katharopoulos 等人（2020）；Liu 等人（2022b），我们观察到，将注意力层置于残差块之后能够带来更好的全局连贯性。因此，我们在所有任务中都采用后置注意力机制。值得注意的是，U-Net 对所使用的具体注意力机制并不敏感，这可能是因为注意力层位于较深的层中，此时上下文大小已经足够小，能够轻松捕捉全局上下文。

3.1.4 下采样和上采样层

为了对输入特征进行下采样，我们使用一个步幅为 1×2、填充为0×1的单一 1×3卷积层。这使输入特征的时间维度有效缩小为原来的二分之一，同时保留其他维度不变。

对于上采样，我们使用一个步幅为 1×2、填充为 0×1 的单层 1×4 转置卷积层。这实际上会将输入特征的时间维度扩大 2 倍，同时保留其他维度。

3.2 神经声码器

我们的神经声码器设计受到 ISTFTNet 的启发 Kaneko 等人(2022)，并采用了简单的结构。它接收输入梅尔频谱图 $\mathbf{X} \in \mathbb{R}^{c \times f_m \times l}$，其中 f_m 是梅尔频谱图的频率维度，并将其传入输入层（见第 3.1.1节）。随后，梅尔频谱图经过一个残差块（见第 3.1.2节），不使用时间步嵌入特征，最后传入输出层（见第 3.1.1节）。由此得到 STFT 频谱图 $\mathbf{Y} \in \mathbb{R}^{c \times f_s \times l}$，其中$f_s$ 是幅度 STFT 频谱图的频率维度。经过输出层后，我们应用指数激活，将幅度频谱图从对数空间转换到线性空间（见图 1）。

4 实验

4.1 数据集

本研究使用的数据集是 POP909 数据集Wang* 等人 (2020)，其中包含 909 个流行歌曲的 MIDI 文件。我们使用 FluidSynth 将 MIDI 文件合成为 44.1 kHz 立体声音频 Newmarch (2017)。

4.2 训练细节

4.2.1 数据预处理

为了从原始音频合成梅尔频谱图，我们使用窗长为 2048、跳步长度为 1024 的 STFT，并对生成的梅尔频谱图应用 128 个梅尔滤波器组。由于扩散模型通常处理范围为 $[-1, 1]$ 内的数据，并且要求数据服从高斯分布，我们提出学习数据专用预处理技术，例如用于标准缩放的移动平均参数（参见算法 1）和最小-最大缩放（详情请参阅第 A 节）。

4.2.2 模型训练

对于我们的 U-Net 模型，我们将宽度设为 256，并在应用下采样/

上采样层之前，每个分辨率使用 2 个 U-Net 块。这样，编码器和解码器中总共有 14 个 U-Net 块，模型包含 4980 万个参数。我们将时间步维度设为 128，并使用 Adam 优化器训练模型 110,000 步，其中包含 500 步的线性预热。Adam 优化器的 β_1 值设为 0.5，学习率设为 0.0002。我们还训练了 U-Net 的指数移动平均版本。音频长度限制为 8,387,584 个采样点（ 190 秒），这样生成的频谱图便可被下采样层的数量整除。我们使用批次大小为 4，并在一块内存为 16 GB 的 GPU 上训练模型。具体使用哪款 GPU 可能取决于设备可用情况。

对于神经声码器模型，我们将宽度设为 256，并在输入层和输出层之间使用一个残差块。由此得到的模型包含 140 万个参数。我们使用 Adam 优化器，以 500 步的线性预热训练模型 40,000 步。Adam 优化器的 β_1 值设为 0.5，学习率设为 0.0002。音频长度限制为 523,264 个采样点（11 秒），这样生成的频谱图就能被下采样层的数量整除。我们使用 8 的批次大小，并在一块内存为 16 GB 的 GPU 上训练模型。具体使用哪种 GPU 可能因可用情况而异。

我们的两个模型均采用 16 位浮点精度进行训练。

对于扩散，我们使用 diffusers 库von Platen 等人（2022）对 DDIM 算法的实现Song 等人（2020），设置 1000 个时间步，并采用 Glide 的余弦噪声调度 Nichol 等人（2021）（更多详情请参见第B 节）。

4.3 评估

我们的模型目前仍在积极开发中，因此我们对样本进行了人工评估。在此次评估中，我们重点关注生成样本的长期连贯性和和声。我们使用不同的随机种子随机生成样本，而不是挑选特定样本。不过，我们尚未实现任何用于评估模型性能的定量指标。

5 结果

5.1 采样

为了评估生成样本的质量，我们采用人类听众进行主观评估。我们生成的样本使用 200 步 DDIM 和 200 次 GriffinLim 迭代。

总体而言，我们观察到生成的样本具有良好的长期连贯性，能够维持约 3 分钟的连贯性。样本还呈现出多样的结构，包括贯穿整首 Song 的重复模式。然而，与人类创作的音乐相比，生成的样本质量确实有所下降，尤其是在真实感和自然度方面。这可能部分归因于使用 GriffinLim 进行相位重建。

这些生成样本一个尤为显著的优势是其多样性，尽管模型是在相对较小的数据集上训练的。这表明模型能够从训练数据中学习到具有泛化能力的模式。

我们还观察到，模型在训练初期难以保持全局连贯性，而且随着训练的进行，损失并未显著改善。然而，我们确实发现，训练时间越长，模型越能实现更好的全局连贯性，整体样本质量也会提升。

我们没有观察到增加 DDIM 采样步数能带来显著改善。这表明，使用更短的噪声调度训练模型，或许可以加快采样速度。要确认这一可能性，还需要进一步实验。

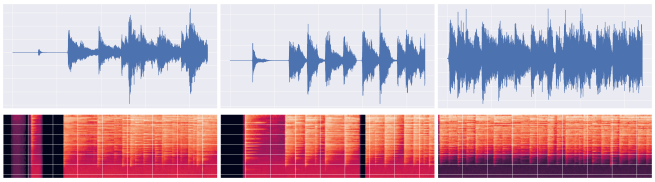

图2：从左到右展示的是使用 200 个 DDIM 步骤和 200 个 GriffinLim 步骤生成的样本。

5.2 音频到音频（风格迁移）

从梅尔频谱图开始 x_0，我们使用公式（6）在所需时间步添加噪声 t，从而得到加噪后的梅尔频谱图 x_t。随后，我们采用反向扩散过程（见第 2.2.2 小节），生成原始音频的变体，使其更接近我们的训练数据。

我们的研究结果表明，在低噪声水平下，生成音频保留了原始音频的结构，同时调整乐器和人声，使其与训练数据中的相匹配。不过，在这些低噪声水平下，生成音频的噪声过大。另一方面，当

噪声水平较高时，生成音频与训练数据更为相似，但原始音频的结构基本丢失。

我们还注意到，鼓等打击乐声对噪声水平升高不太敏感。即使噪声水平很高，生成音频的整体结构仍能保持。

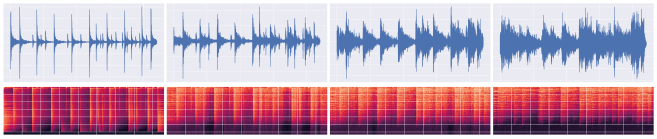

图 3：从左到右展示的是鼓循环的音频到音频样本，其噪声时间步依次递增：t:$t = 0$、$t = 100$、$t = 500$ 和 $t = 900$。

5.3 插值

使用扩散模型进行音频插值时，我们从两个梅尔频谱图开始 $\boldsymbol{x}_0^1$ 和 $\boldsymbol{x}_0^2$，并使用公式（6）在所需时间步 t添加噪声，从而得到加噪后的梅尔频谱图 $\boldsymbol{x}_t^1$ 和 $\boldsymbol{x}_t^2$。接着，我们按如下方式对这两个加噪后的频谱图进行插值：$\boldsymbol{x}_t = \gamma \boldsymbol{x}_t^1 + (1-\gamma)\boldsymbol{x}_t^2$，其中 $\gamma \in [0,1]$ 是插值比例。我们应用反向扩散过程（见第 2.2.2小节），生成两个音频源的插值结果，使其更接近我们的训练数据。

我们发现，即使插值比例较低，生成音频中的打击乐声往往也更为突出。与音频到音频任务的结果类似（参见第 5.2 小节），我们观察到，在噪声水平较低时，原始音频的音乐结构得以保留，但生成音频带有噪声。另一方面，在噪声水平较高时，生成音频的音乐结构与训练数据的结构非常相似。

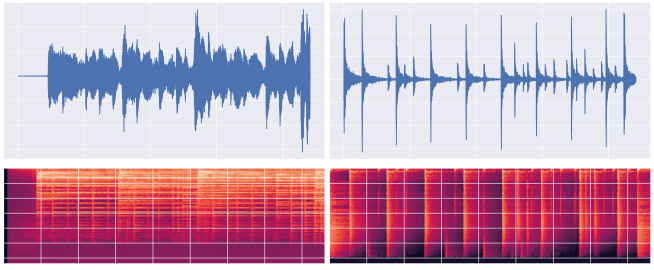

图4：用于插值的原始样本，左侧为钢琴样本，右侧为鼓循环。

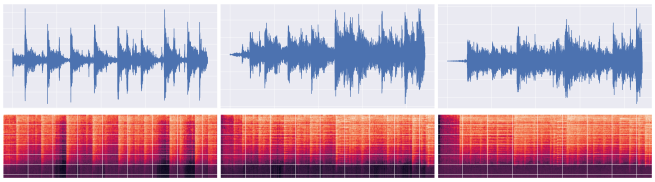

图5：从左至右依次展示使用固定噪声时间步 $t = 200$生成的插值样本，其原始样本比例分别为 0.1、0.5 和 0.9，表示每个插值样本中第一个原始样本所占的比例。

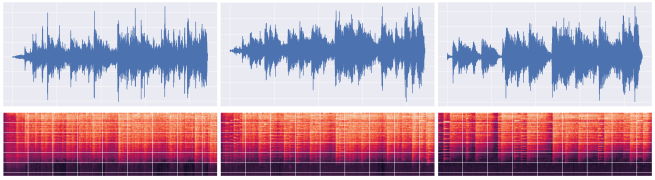

图6：从左至右依次展示使用固定比例 0.5、噪声时间步 t 逐渐增大的插值样本：$t = 100$、$t = 500$和 $t = 900$。

5.4 内绘

在内绘中，我们使用二值掩码来标识音频信号中需要保留和需要移除的部分。随后，我们应用 Lugmayr等人（2022）提出、由 von Platen等人（2022）实现的 RePaint算法，填补掩码区域。

然而，我们的结果表明，修复片段往往缺乏节奏，无法准确捕捉原始音频的旋律和和声。修复后的音频听起来像一个全新的样本，完全不像原始音频。这导致歌曲的修复片段中音乐结构发生突然变化。

我们不确定此问题为何会发生。还需要进一步实验来调查并解决这一问题。

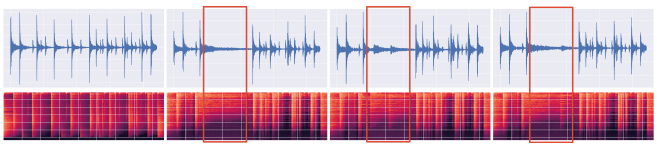

图 7：从左至右依次为：原始鼓循环样本，以及使用不同随机种子生成的三个修复版本。每个样本中修复的部分均以红框标出。

5.5 外绘

外绘是指通过填充合成音频片段，将音频延伸至原始录音之外。进行外绘时，我们使用与内绘相同的算法（参见第5.4小节）。我们取一半原始音频，并将其与空频谱图拼接，然后指定一个掩码，对空白部分进行内绘。

遗憾的是，我们的结果并不理想。外绘部分听起来往往与原始音频不同，而且经常缺乏节奏。这种情况尤为明显，因为外绘的音频片段不同，导致音频中出现突变区域，使其听起来像是多个不同音源拼接在一起。

我们不确定这一问题为何会发生。还需要进一步实验来调查并解决这一问题。

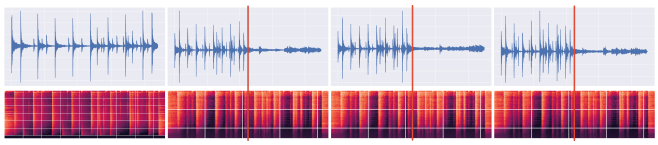

图 8：从左到右依次为一个原始鼓循环样本，以及使用不同随机种子生成的三个外绘版本。每个样本中外绘部分的起始位置以红线标出。

6 未来工作

这项工作可以沿着多个方向扩展，以提高该方法在音乐制作场景中的质量和实用性。未来研究的一些潜在方向包括：

6.1 条件生成

在这项工作中，我们专注于无条件生成，也就是说，几乎无法控制合成的样本。然而，在音乐制作场景中，允许用户对生成音乐进行一定程度的控制十分重要。为合成过程增加可控性的一些可能方法包括：

- 根据歌词、情绪或 MIDI 输入等因素对模型进行条件设定。这样用户就可以指定生成音乐的内容或风格。
- 允许用户在合成过程中提供反馈，无论是通过明确输入，还是通过与生成音频交互。这可以让用户实时塑造输出，或引导合成过程，以获得他们期望的结果。

6.2 现实场景中的评估

我们的模型尚未在真实场景中进行评估，因此有必要确定它在音乐制作中的实用性和有效性。评估该模型的可能方法包括：

- 在音乐制作场景中开展用户研究或案例研究，以评估这些技术的实用性和有效性。这可能包括收集音乐制作人、音乐家或其他行业专业人士的反馈。
- 将模型的输出与人类音乐家的作品或现有音乐制作工具的输出进行比较，采用主观质量或特定任务表现等指标。这有助于发现模型相较于人类或既定标准的优势与不足。

6.3 推广至其他音频任务

尽管本研究聚焦于音乐合成，我们相信该模型也有望应用于其他音频处理任务。值得探索的领域包括：

- 分类任务，例如流派分类或乐器识别。这有助于确定模型学习和表征音乐概念的能力。
- 音频修复任务，例如噪声降低或音频增强。这有助于评估模型操控和改善音频信号的能力。

6.4 对 ModelComponents 的改进

可以通过多种方式提升模型及其组件的性能：

- 解决内绘和外绘在交互式音频设计中的性能欠佳问题。这可能包括重新设计内绘算法，或采用新的音频合成技术。
- 扩大模型规模以提升其性能。这有助于评估模型的缩放定律。
- 融入扩散模型采样领域的新进展，实现近乎实时的合成。这有助于提升模型在音乐制作环境中的响应速度和交互性。

6.5 扩展生成音乐的范围

在这项工作中，我们专注于生成单一乐器的音乐。不过，该模型还可以通过多种方式扩展，以生成种类更丰富的音乐：

- 考察模型如何处理包含多种乐器和人声的更复杂数据。这或许能让模型生成范围更广的音乐风格和织体。
- 探索多轨或分轨生成的用法，让用户能够明确控制每种生成的乐器。这在音乐制作场景中尤其有用。
- 研究使用该模型生成不同风格或流派音乐的方式，例如电子、古典或世界音乐。这有助于评估模型学习和呈现多样音乐传统的能力。

总体而言，我们的方法及其在音乐合成人工智能领域的应用具有巨大的进一步研究潜力；我们相信，这些技术有望对该领域产生重大影响。

7 结论

在这项工作中，我们提出了 Msanii，这是一种新颖的扩散模型，可高效合成长上下文、高保真音乐。通过结合梅尔频谱图的表现力、扩散模型的生成能力以及神经声码器的声码合成能力，Msanii 能够生成高质量音频。我们的结果展示了 Msanii 高效生成数分钟连贯音频的能力；我们也探讨了 Msanii 在音乐制作领域各种应用中的潜力。凭借出色的性能和多功能性，我们相信 Msanii 具有巨大的进一步研究与开发潜力。总体而言，Msanii 有望成为音乐合成与制作的强大工具。

8 致谢

我们谨向 OpenAI 致以谢意，感谢其提供 ChatGPT 的使用权限，这对修改我们的论文起到了重要作用。ChatGPT 的使用显著提升了我们论文的清晰度和可读性，我们也感谢它所提供的帮助。

参考文献

Arık, S. Ö., Jun, H., and Diamos, G. (2018). 使用多头卷积神经网络进行快速频谱图反演。*IEEE*信号处理快报，26(1):94–98.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., 等（2020）。语言模型是少样本学习者。神经信息处理系统进展，33：1877–1901。

Dhariwal, P.、Jun, H.、Payne, C.、Kim, J. W.、Radford, A. 和 Sutskever, I.（2020）。Jukebox：用于音乐的生成模型。*arXiv* 预印本 *arXiv:2005.00341*。

Dhariwal, P. 和 Nichol, A. (2021)。扩散模型在图像合成方面胜过生成对抗网络。神经信息处理系统进展，34:8780–8794。Di Giorgi, B.、Levy, M. 和 Sharp, R. (2022)。利用稳定音高进行梅尔频谱图反演。*arXiv* 预印本 *arXiv:2208.12782*。Dieleman, S.、van den Oord, A. 和 Simonyan, K. (2018)。逼真音乐生成的挑战：大规模建模原始音频。神经信息处理系统进展，31。Dosovitskiy, A.、Beyer, L.、Kolesnikov, A.、Weissenborn, D.、Zhai, X.、Unterthiner, T.、Dehghani, M.、Minderer, M.、Heigold, G.、Gelly, S. 等 (2020)。一张图像胜过 16×16 个词：面向大规模图像识别的 Transformer。*arXiv* 预印本 *arXiv:2010.11929*。Engel, J.、Agrawal, K. K.、Chen, S.、Gulrajani, I.、Donahue, C. 和 Roberts, A. (2019)。GANSynth：对抗性神经音频合成。*arXiv* 预印本 *arXiv:1902.08710*。Goodfellow, I.、Pouget-Abadie, J.、Mirza, M.、Xu, B.、Warde-Farley, D.、Ozair, S.、Courville, A. 和 Bengio, Y. (2020)。生成对抗网络。美国计算机协会通讯，63(11):139–144。Griffin, D. 和 Lim, J. (1984)。从修正短时傅里叶变换进行信号估计。*IEEE*声学、语音与信号处理汇刊，32(2):236–243。Ho, J.、Jain, A. 和 Abbeel, P. (2020)。去噪扩散概率模型。神经信息处理系统进展，33:6840–6851。Hwang, Y.、Cho, H.、Yang, H.、Won, D.-O.、Oh, I. 和 Lee, S.-W. (2020)。用于序列到序列语音转换的梅尔频谱图增强。*arXiv* 预印本 *arXiv:2001.01401*。

Kaneko, T.、Tanaka, K.、Kameoka, H. 和 Seki, S. (2022)。ISTFTNet：结合逆短时傅里叶变换的快速轻量级梅尔频谱声码器。载于*ICASSP 2022-2022 IEEE*国际声学、语音与信号处理会议（*ICASSP*），第 6207–6211 页。IEEE。

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J. 和 Aila, T. (2020). 分析并改进 StyleGAN 的图像质量。载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，第 8110–8119 页。

Katharopoulos, A., Vyas, A., Pappas, N. 和 Fleuret, F. (2020). Transformer 就是循环神经网络：具有线性注意力的快速自回归 Transformer。载于国际机器学习会议，第 5156–5165 页。PMLR。

Kingma, D., Salimans, T., Poole, B. 和 Ho, J. (2021). 变分扩散模型。神经信息处理系统进展，34:21696–21707。

Kingma, D. P. 和 Welling, M.（2013）。自编码变分贝叶斯。*arXiv*预印本 *arXiv:1312.6114*。

Kong, Z.、Ping, W.、Huang, J.、Zhao, K. 和 Catanzaro, B.（2020）。DiffWave：用于音频合成的多功能扩散模型。*arXiv* 预印本 *arXiv:2009.09761*。

Kumar, K.、Kumar, R.、de Boissiere, T.、Gestin, L.、Teoh, W. Z.、Sotelo, J.、de Brébisson, A.、Bengio, Y. 和 Courville, A. C.（2019）。MelGAN：用于条件波形合成的生成对抗网络。神经信息处理系统进展，32。

Liu, J.、Li, C.、Ren, Y.、Chen, F. 和 Zhao, Z.（2022a）。DiffSinger：通过浅层扩散机制实现的歌声合成。载于*AAAI*人工智能会议论文集，第 36 卷，第 11020–11028 页。

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., 和 Xie, S.（2022b）。面向 2020 年代的卷积网络。载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，第 11976–11986 页。

Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., 和 Van Gool, L.（2022）。RePaint：使用去噪扩散概率模型进行内绘。载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，第 11461–11471 页。

Luo, C.（2022）。理解扩散模型：统一视角。*arXiv* 预印本 *arXiv:2208.11970*。

Marafioti, A.、Perraudin, N.、Holighaus, N. 和 Majdak, P.（2019）。时频特征的对抗生成 features w ith application in audi合成 hesis. In *I* 国际机器学习会议，第 4352–4362 页。PMLR。

Newmarch, J.（2017）。FluidSynth。载于*Linux* 声音编程，第 351–353 页。施普林格。Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., 和 Chen, M.（2021）。Glide：迈向借助文本引导扩散模型实现逼真图像生成与编辑。*arXiv* 预印本 *arXiv:2112.10741*。

Nichol, A. Q. 和 Dhariwal, P.（2021）。改进的去噪扩散概率模型。载于国际机器学习会议，第 8162–8171 页。PMLR。

Pasini, M. 和 Schlüter, J. (2022)。Musika!：快速、无限时长的波形音乐生成。*arXiv* 预印本 *arXiv:2208.08706*。

Perraudin, N.、Balazs, P. 和 Søndergaard, P. L. (2013)。一种快速的 Griffin-Lim 算法。载于*2013 IEEE* 音频与声学信号处理应用研讨会，第 1–4 页。IEEE。

Ramesh, A.、Dhariwal, P.、Nichol, A.、Chu, C. 和 Chen, M. (2022)。基于 CLIP 潜变量的分层文本条件图像生成。*arXiv* 预印本 *arXiv:2204.06125*。Rezende, D. 和 Mohamed, S. (2015)。利用归一化流进行变分推断。载于国际机器学习大会，第 1530–1538 页。PMLR。

Rombach, R.、Blattmann, A.、Lorenz, D.、Esser, P. 和 Ommer, B. (2022)。利用潜扩散模型进行高分辨率图像合成。载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，第 10684–10695 页。Ronneberger, O.、Fischer, P. 和 Brox, T. (2015)。U-Net：用于生物医学图像分割的卷积网络。载于国际医学图像计算与计算机辅助手术会议，第 234–241 页。施普林格。Rouard, S. 和 Hadjeres, G. (2021)。CRASH：用于可控高分辨率鼓声合成的原始音频基于分数的生成式建模。*arXiv* 预印本 *arXiv:2106.07431*。Saharia, C.、Chan, W.、Saxena, S.、Li, L.、Whang, J.、Denton, E.、Ghasemipour, S. K. S.、Ayan, B. K.、Mahdavi, S. S.、Lopes, R. G. 等 (2022)。借助深度语言理解实现的照片级真实感文本到图像扩散模型。*arXiv* 预印本 *arXiv:2205.11487*。

Scao, T. L.、Fan, A.、Akiki, C.、Pavlick, E.、Ili´c, S.、Hesslow, D.、Castagné, R.、Luccioni, A. S.、Yvon, F.、Gallé, M. 等 (2022)。Bloom：一个拥有 1760 亿参数的开放访问多语言语言模型。*arXiv* 预印本 *arXiv:2211.05100*。Shen, J.、Pang, R.、Weiss, R. J.、Schuster, M.、Jaitly, N.、Yang, Z.、Chen, Z.、Zhang, Y.、Wang, Y.、Skerry-Ryan, R. 等 (2017)。通过以梅尔频谱图预测为条件的 WaveNet 实现自然语音合成。arXiv 电子印本。*arXiv* 预印本 *arXiv:1712.05884*。Sohl-Dickstein, J.、Weiss, E.、Maheswaranathan, N. 和 Ganguli, S. (2015)。利用非平衡热力学进行深度无监督学习。载于国际机器学习大会，第 2256–2265 页。PMLR。Song, J.、Meng, C. 和 Ermon, S. (2020)。去噪扩散隐式模型。*arXiv* 预印本 *arXiv:2010.02502*。Ulyanov, D.、Vedaldi, A. 和 Lempitsky, V. (2016)。实例归一化：快速风格化不可或缺的要素。*arXiv* 预印本 *arXiv:1607.08022*。Vasquez, S. 和 Lewis, M. (2019)。MelNet：频域音频生成模型。*arXiv* 预印本 *arXiv:1906.01083*。Vaswani, A.、Shazeer, N.、Parmar, N.、Uszkoreit, J.、Jones, L.、Gomez, A. N.、Kaiser, Ł. 和 Polosukhin, I. (2017)。注意力就是你所需的一切。神经信息处理系统进展，第 30 卷。von Platen, P.、Patil, S.、Lozhkov, A.、Cuenca, P.、Lambert, N.、Rasul, K.、Davaadorj, M. 和 Wolf, T. (2022)。Diffusers：最先进的扩散模型。https://github.com/huggingface/diffusers。Wang*、Z.、Chen*、K.、Jiang, J.、Zhang, Y.、Xu, M.、Dai, S.、Bin, G. 和 Xia, G. (2020)。Pop909：用于音乐编曲生成的流行歌曲数据集。载于第 *21* 届国际音乐信息检索大会论文集，*ISMIR*。Weng, L. (2021)。什么是扩散模型？*lilian-weng.github.io*。

数据预处理

A.1 标准缩放（归一化）

算法 1 标准缩放算法

输入： x：小批次 $B = \{x_{1...n}\}$，m：动量，d：动量衰减，ϵ：艾普西隆

输出： $y = \frac{x - \mathbb{E}[x]}{\sqrt{\text{Var}[x] + \epsilon}}$

if 没有运行统计量 **then**
  # 初始化运行统计量
  $\mu_R \leftarrow \frac{1}{n}\sum_{i=1}^{n} x_i$ # 均值
  $\sigma_R^2 \leftarrow \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_R)^2$ # 方差
结束 if

if 处于训练模式 **then**
  # 使用小批量统计
  $\mu_B \leftarrow \frac{1}{n}\sum_{i=1}^{n} x_i$ # 均值
  $\sigma_B^2 \leftarrow \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_B)^2$ # 方差

  # 更新运行统计量
  $m \leftarrow d \cdot m$ # 动量
  $\mu_R \leftarrow (1 - m) \cdot \mu_R + m \cdot \mu_B$ # 均值
  $\sigma_R^2 \leftarrow (1 - m) \cdot \sigma_R^2 + m \cdot \sigma_B^2$ # 方差
否则
  # 使用运行统计量
  $\mu_B \leftarrow \mu_R$ # 均值
  $\sigma_B^2 \leftarrow \mu_R$ # 方差
结束 if

$y \leftarrow \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$

此算法使用输入的均值 $\mathbb{E}[x]$ 和方差 $\text{Var}[x]$ 对输入 x的各个元素进行标准化。训练期间，使用动量 m 和动量衰减 d 更新运行统计量，并使用小常数 ϵ 来防止除以零。如果模型处于训练模式，则使用批次统计量 μ_B 和 σ_B^2 来计算均值和方差。如果模型处于推理模式，则改为使用运行统计量 μ_R 和 σ_R^2。

A.2 最小-最大缩放

算法 2 最小-最大缩放算法

输入： x：小批次 $B = \{x_{1...n}\}$，y_{min}：最小值缩放后，y_{max}：缩放后的最大值，m：动量，d：动量衰减

输出： $y = \frac{x - \text{最小值}(x)}{\text{最大值}(x) - \text{最小值}(x)} * (y_{max} - y_{min}) + y_{min}$

if 没有运行统计量 **then**
  # 初始化运行统计量
  $\text{最小值}_R \leftarrow \text{最小值}(x)$ # 最小值
  $\text{最大值}_R \leftarrow \text{最大值}(x)$ # 最大值
结束 if

if 处于训练模式 **then**
  # 使用小批量统计
  $\text{最小值}_B \leftarrow \text{最小值}(x)$ # 最小值
  $\text{最大值}_B \leftarrow \text{最大值}(x)$ # 最大值

  # 更新运行统计量
  $m \leftarrow d \cdot m$ # 动量
  $\text{最小值}_R \leftarrow (1 - m) \cdot \text{最小值}_R + m \cdot \text{最小值}_B$ # 最小值
  $\text{最大值}_R \leftarrow (1 - m) \cdot \text{最大值}_R + m \cdot \text{最大值}_B$ # 最大值
否则
  # 使用运行统计量
  $\text{最小值}_B \leftarrow \text{最小值}_R$
$\text{最大值}_B \leftarrow \text{最大值}_R$
结束 if

$y = \frac{x - \text{最小值}_B}{\text{最大值}_B - \text{最小值}_B} * (y_{max} - y_{min}) + y_{min}$
$y = \max(y_{min}, \min(y, y_{max}))$ # 将 y 限制在范围 $[y_{min}, y_{max}]$

该算法会缩放输入 x 的元素，使其处于范围 $[y_{min}, y_{max}]$ 内。训练期间，动量 m 和动量衰减 d 用于更新运行统计量。如果模型处于训练模式，则使用批次统计量 $\min_B$ 和 $\max_B$ 计算最小值和最大值。如果模型处于推理模式，则改用运行统计量 $\min_R$ 和 $\max_R$。

B 超参数

B.1 神经声码器

超参数	值
Data	
采样率	44100
音频长度	523264
音频通道	2
批次大小	8
变换	
快速傅里叶变换大小	2048
窗长	2048
跳长	1024
梅尔频率	128
特征缩放动量	0.001
特征缩放衰减	0.99
Griffin-Lim 迭代次数	200
声码器	
模型维度	256
MLP 隐藏维度因子	4
训练	
学习率（lr）	0.0002
优化器	Adam
Adam 的 β 参数	0.5, 0.999
学习率预热迭代次数	500
学习率预热起始因子	$\frac{1}{3}$
精度	16
训练步数	40000

表 1：神经声码器超参数。

B.2 U-Net

超参数	值
Data	
采样率	44100
音频长度	8387584
音频通道	2
批次大小	4
扩散	
噪声调度	余弦
训练时间步数	1000
采样步数	200
U-Net	
基础模型维度	256
时间步维度	128
MLP 隐藏维度因子	4
注意力头数	8
维度因子	1,1,1,1,1,1,1
膨胀	1 1 1 1 1 1 1
包含注意力	F, F, F, F, F, T, T
包含重采样	T、T、T、T、T、T、F
每个分辨率的块数	2,2,2,2,2,2,2
EMA U-Net	
起始步数	2000
衰减	0.995
每 n 步更新	10
训练	
学习率（lr）	0.0002
优化器	Adam
Adam 的 β 参数	0.5, 0.999
学习率预热迭代次数	500
学习率预热起始因子	$\frac{1}{3}$
精度	16
训练步数	110000

表 2：U-Net 超参数。

C 优化 U-Net 宽度以提升性能

U-Net 架构与 Transformer 类似，通过增加宽度可以显著提升性能。然而，使用 U-Net 作为特征时，必须考虑 U-Net 的宽度与频谱图频率维度之间的关系。

例如，使用宽度低于频率维度的频谱图 U-Net，会迫使输入层只能学习梅尔频谱图的主成分。这种方法对于干净的梅尔频谱图可能效果良好，但当梅尔频谱图受到噪声干扰时则并不适用。

我们的实验表明，要确保最佳性能，U-Net 宽度至少应为频谱图频率维度的 2 倍。还需要进一步实验，才能充分理解这一现象，并优化 U-Net 宽度以提升性能。