arXiv:2304.00830v2 [cs.SD] 5 Apr 2023

AUDIT：使用潜在扩散模型遵循指令进行音频编辑

Yuancheng Wang[1 2*] **Zeqian Ju**[1], **Xu Tan**[1], **Lei He**[1], **Zhizheng Wu**[2], **Jiang Bian**[1], **Sheng Zhao**[11]微软， [2]香港中文大学（深圳）[1]{v-yuancwang,v-zeqianju,xuta,helei,jiabia,szhao}@microsoft.com[2]yuanchengwang@link.cuhk.edu.cn,wuzhizheng@cuhk.edu.cn

摘要

音频编辑可用于多种目的，例如添加背景音效、替换乐器以及修复受损音频。最近，一些基于扩散的方法通过以输出音频的文本描述为条件进行扩散与去噪，实现了零样本音频编辑。然而，这些方法仍存在一些问题：1）它们未在编辑任务上接受训练，因此无法保证良好的编辑效果；2）它们可能会错误地修改无需编辑的音频片段；3）它们需要对输出音频进行完整描述，而在实际场景中，这种描述并不总是可用或必要。在这项工作中，我们提出了**AUDIT**，一种基于潜在扩散模型的指令引导音频编辑模型。具体而言，AUDIT 具有三个主要设计特点：1）我们针对不同的音频编辑任务构建三元组训练数据（指令、输入音频、输出音频），并训练一个扩散模型，以指令和待编辑的输入音频为条件，生成编辑后的输出音频；2）通过比较输入音频与输出音频之间的差异，它能够自动学习只修改需要编辑的片段；3）作为文本输入，它只需要编辑指令，而不需要完整的目标音频描述。对于多种音频编辑任务（例如添加、删除、替换、图像修复和超分辨率），AUDIT 在客观和主观指标上均取得了最先进的结果。演示样本可在 https://audit-demo.github.io/ 获取。

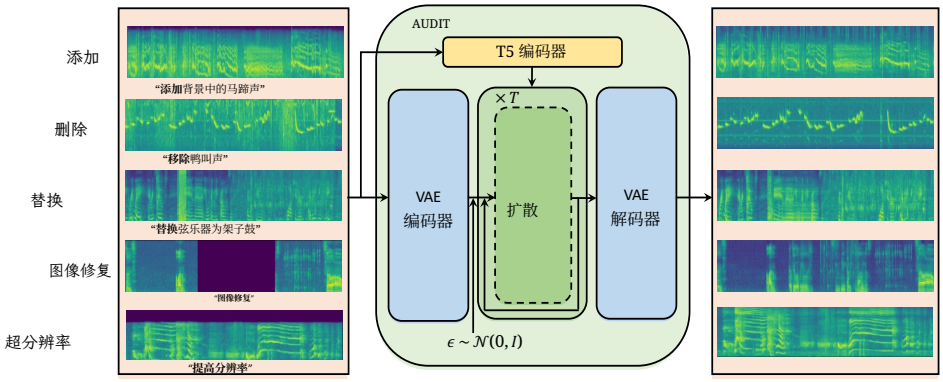


图 1：AUDIT 由变分自编码器、T5文本编码器和扩散网络组成，并将输入音频的梅尔频谱图和编辑指令作为条件输入，生成编辑后的音频作为输出。

*在微软实习期间完成的工作。

预印本。审稿中。

1 引言

音频编辑是指对输入音频执行编辑操作（例如添加、删除、替换、图像修复和超分辨率），以获得编辑后的音频。它有多种应用场景，例如添加背景音效、替换背景音乐、修复不完整的音频、增强低采样率音频等。能够遵循人类指令的音频编辑工具，可以让编辑更加灵活，并符合人们的预期。用户可以通过“在背景中添加爵士乐”“删除鸟叫声”或“进行图像修复”等指令，轻松地按需编辑音频。在这项工作中，我们专注于开发能够遵循人类指令的音频编辑模型。

以往的音频编辑工作通常采用传统信号处理技术。后来，基于生成对抗网络的方法 [32, 10, 1] 在音频修复和超分辨率等一些音频编辑任务中取得了成功。然而，这些方法通常是为特定的编辑任务而设计的。近年来，最先进的深度生成模型——扩散模型——在图像编辑领域取得了巨大成功。一些方法 [18, 27] 尝试将扩散模型应用于音频编辑领域。这些基于扩散模型的音频编辑方法主要依赖于预训练的文本到音频生成模型。然而，这些方法仍存在若干问题：1）它们主要依赖预训练模型，在目标文本描述的条件下对输入音频进行加噪和去噪；由于大多数情况下并未直接针对音频编辑任务进行训练，因此无法保证获得良好的编辑效果；2）由于难以限制编辑区域，它们可能会错误地修改一些无需编辑的音频片段；3）它们要求完整描述输出音频，而在真实世界的编辑场景中，这类描述并非总是可用或必要。相较之下，对于音频编辑，理想的做法是提供自然语言编辑指令，例如“在背景中添加一个男人吹口哨”或“去掉吉他的声音”。

为解决上述问题，据我们所知，我们提出了 AUDIT，这是首个基于人类指令的音频编辑模型。如图 1 所示，AUDIT 是一种潜在扩散模型，以待编辑的音频和编辑指令作为条件，并生成编辑后的音频作为输出。AUDIT 的核心设计可概括为三点：1）我们为不同的音频编辑任务生成三元组数据（指令、输入音频、输出音频），并以监督方式训练音频编辑扩散模型；2）我们直接将输入音频用作条件输入来训练模型，促使模型自动学习如何确保无需编辑的音频片段在编辑前后保持一致；3）我们的方法直接使用编辑指令作为文本引导，无需完整描述输出音频，因此更加灵活，也更适用于现实场景。我们的编辑模型在客观和主观指标上均取得了最先进的结果。

我们的贡献可总结如下：

• 我们提出了 AUDIT，这是首次尝试训练一种以人工文本指令为条件、用于音频编辑的潜在扩散模型。

• 为了以监督方式训练 AUDIT，我们为每项编辑任务设计了一种新颖的数据构建框架；AUDIT 能最大限度地保留无需编辑的音频片段；AUDIT 只需简单的指令作为文本引导，无需完整描述编辑目标。

• 在若干音频编辑任务中，AUDIT 的客观和主观评价指标均达到最先进水平。

2 相关工作

2.1 音频/语音编辑

以往与音频编辑相关的工作主要聚焦于人类语音或音乐。 [35, 5, 32, 52]探索了基于对应文本或乐谱的语音/音乐修复任务。另一类编辑工作则聚焦于语音或音乐的风格迁移。语音转换 [2, 34, 42, 41] 和歌声转换 [7, 28] 旨在不改变语音或歌唱内容的情况下修改音色。音乐风格迁移 [48, 9, 30, 38] 旨在改变演奏的乐器，而不修改乐谱。此外，一些使用生成对抗网络 [32, 10, 1]的方法已在某些特定音频编辑任务中取得成功，例如音频修复和音频

超分辨率。然而，针对遵循人类指令的通用音频数据的研究仍然不足。

2.2 基于扩散的编辑

基于扩散的编辑任务与基于扩散的生成任务一样受到了广泛关注。我们将常见的基于扩散的编辑工作归纳为两类。

零样本编辑一些方法 [33, 8, 4, 31, 14] 使用基于扩散的生成模型实现零样本编辑。SDEdit [33]使用预训练扩散模型对输入图像添加噪声，然后使用新的目标文本对图像进行去噪。这些方法难以编辑特定区域，且可控性较差。为编辑特定区域， [4]使用额外的损失梯度来引导采样过程； [4, 31] 则在每个采样步骤中用带噪的输入图像替换未修改的部分，这相当于仅在掩码部分进行生成。然而，这些方法仅适用于修复任务，而且需要对编辑目标进行完整描述。

监督训练另一类方法 [45, 6, 51, 36]以监督方式处理编辑任务。 [45]训练一种以待编辑图像为条件的图像到图像扩散模型。 [45] 使用由 [14] 生成的配对图像数据来训练基于文本的图像编辑模型。ControlNet [51] 和 T2I-Adapter [36] 在冻结的预训练生成扩散模型中加入适配器模型，并通过训练该适配器模型实现图像编辑。我们的方法遵循监督训练范式，使用生成的三元组数据（指令、输入音频、输出音频）训练以指令和待编辑音频为条件的音频编辑模型，并能够根据人类指令处理不同的音频编辑任务。

2.3 文本引导的音频生成

目前，大多数基于扩散的音频编辑方法 [18, 27] 都基于文本到音频扩散生成模型。文本到音频生成旨在合成与文本描述相匹配的通用音频。DiffSound [49] 是首个尝试构建基于离散扩散模型 [13]的文本到音频系统的方法。AudioGen [26] 是另一种基于采用自回归结构的基于 Transformer 的解码器的文本到音频生成模型。AudioGen 直接预测从波形 [50]压缩得到的离散标记。近期，Make-an-Audio [18] 和 AudioLDM [27] 尝试在连续空间中构建文本引导的音频生成系统：这些系统使用自编码器将梅尔频谱图转换为潜在表示，在潜在空间中构建扩散模型，并使用自编码器将潜在扩散模型预测的结果重建为梅尔频谱图。此外，MusicLM [3] 和 Noise2Music[17] 会生成与输入文本中的语义信息相匹配的音乐。我们的工作则侧重于根据人类指令编辑现有音频。

3 方法

本节介绍我们基于人工指令实现音频编辑的方法。我们在第3.1节概述系统架构和训练目标，并在第3.2节详细说明如何为每个编辑任务生成三元组数据（指令、输入音频、输出音频）。最后，我们在第3.3节讨论该方法的优势。

3.1 系统概述

我们的系统由自编码器、文本编码器、扩散网络和声码器组成。自编码器将输入梅尔频谱图投影到高效的低维潜在空间，并将其重建为梅尔频谱图；文本编码器对输入文本指令进行编码；扩散网络在潜在空间中进行编辑；声码器则用于重建波形。系统概览如图2所示。

自编码器 自编码器包含一个编码器 E 和一个解码器 G。编码器 E 将梅尔频谱图 $\boldsymbol{x}$ 转换为潜在表示 $\boldsymbol{z}$，解码器 G 则从潜在空间重建 $\hat{}\boldsymbol{x}$ 。在我们的工作中，我们采用变分自编码器 (VAE) [23] 模型作为自编码器。我们使用以下损失训练自编码器：1) $\mathcal{L}_1$ 和 $\mathcal{L}_2$ 重建损失。2)

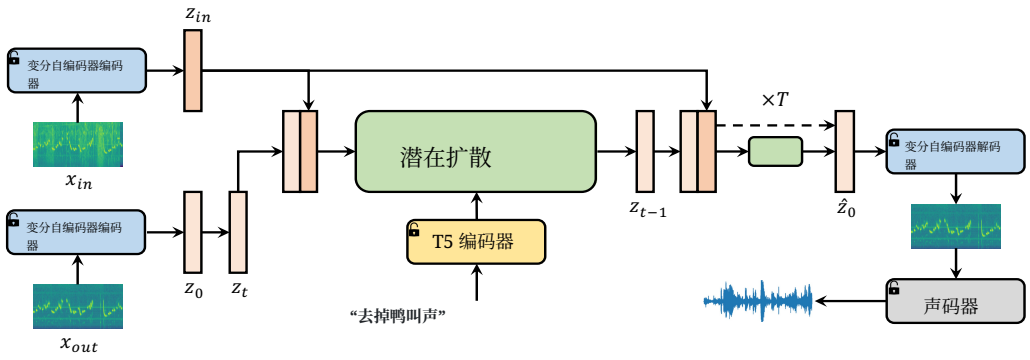


图2：我们的系统概览。

库尔贝克–莱布勒损失 $\mathcal{L}_{KL}$ 用于正则化潜在表示 $\boldsymbol{z}$。按照 [43]的做法，我们仅为 KL损失赋予较小的权重 λ_{KL}；3) 生成对抗网络损失 $\mathcal{L}_{GAN}$。我们采用基于图像块的判别器D [19, 43]，以区分真实梅尔频谱图与重建的梅尔频谱图。自编码器的总训练损失可表示为 $\mathcal{L}_{VAE} = \lambda_1\mathcal{L}_1 + \lambda_2\mathcal{L}_2 + \lambda_{KL}\mathcal{L}_{KL} + \lambda_{GAN}\mathcal{L}_{GAN}$。 λ_1、λ_2、λ_{KL}和 λ_{GAN} 分别是 $\mathcal{L}_1$、 $\mathcal{L}_2$、 $\mathcal{L}_{KL}$和 $\mathcal{L}_{GAN}$ 的权重。

文本编码器 我们使用预训练语言模型 T5 [21]作为文本编码器，将文本输入转换为包含丰富语义信息的嵌入。文本编码器的参数在训练阶段被冻结。

潜在扩散模型 我们的文本引导的音频编辑模型可以视为条件潜在扩散模型，该模型将输入（待编辑）梅尔频谱图的潜在表示 $\boldsymbol{z}_{in}$ 和文本嵌入作为条件。模型旨在学习 $p(\boldsymbol{z}_{out}|\boldsymbol{z}_{in}, c_{text})$，其中 $\boldsymbol{z}_{out}$ 是输出（编辑后）梅尔频谱图的潜在表示 $\boldsymbol{x}_{out}$， c_{text} 是编辑指令的嵌入。给定潜在表示 $\boldsymbol{z}_{in}$ 和文本指令，我们随机选择一个时间步 t 和噪声 $\boldsymbol{\epsilon}$，通过使用噪声调度生成潜在表示 $\boldsymbol{z}_{out}$ 的含噪版本 $\boldsymbol{z}_t$，然后使用扩散网络 $\boldsymbol{\epsilon_\theta}(\boldsymbol{z}_t, t, \boldsymbol{z}_{in}, c_{text})$预测采样的噪声。训练损失为 $\mathcal{L}_{LDM} = \mathbf{E}_{(\boldsymbol{z}_{in},\boldsymbol{z}_{out},text)}\mathbf{E}_{\epsilon\sim\mathcal{N}(0,\boldsymbol{I})}\mathbf{E}_t||\boldsymbol{\epsilon_\theta}(\boldsymbol{z}_t, t, \boldsymbol{z}_{in}, c_{text}) - \boldsymbol{\epsilon}||_2$。与原始标准扩散模型 [16, 46], 类似，我们使用带有交叉注意力机制 [47] 的 U-Net [44] 作为扩散网络。我们的编辑模型通过在通道维度上直接拼接 $\boldsymbol{z}_t$ 和 $\boldsymbol{z}_{in}$，将 $\boldsymbol{z}_{in}$ 作为条件。因此，U-Net 第一层的输入通道数是最后一层输出通道数的两倍。

声码器 需要使用声码器模型将输出梅尔频谱图转换为音频。在本研究中，我们使用 HiFi-GAN [24] 作为声码器，它是语音合成领域应用最广泛的声码器之一。HiFi-GAN 使用多个小型子判别器来处理不同的周期模式。与一些自回归 [37, 20] 或基于流的 [40] 声码器相比，它兼顾了生成质量和推理效率。

3.2 为每个编辑任务生成三元组训练数据

我们使用生成的三元组数据（指令、输入音频、输出音频）来训练文本引导的音频编辑模型。在本研究中，我们聚焦于五种不同的编辑任务，包括添加、删除、替换、图像修复和超分辨率。请注意，我们使用单个编辑模型训练所有编辑任务。图 3 概述了针对不同编辑任务的数据生成工作流。

添加我们从文本-音频数据集中随机选取两个音频片段 A 和 B，然后将 A和 B 组合起来，得到一个新的音频片段 C。我们将 A 作为输入音频，将 C 作为输出音频，并将 B 的描述文本（或标签）填入指令模板，以获得指令。例如，如图 3 所示，指令模板为“在背景中添加 *{}*”，而 B的描述文本

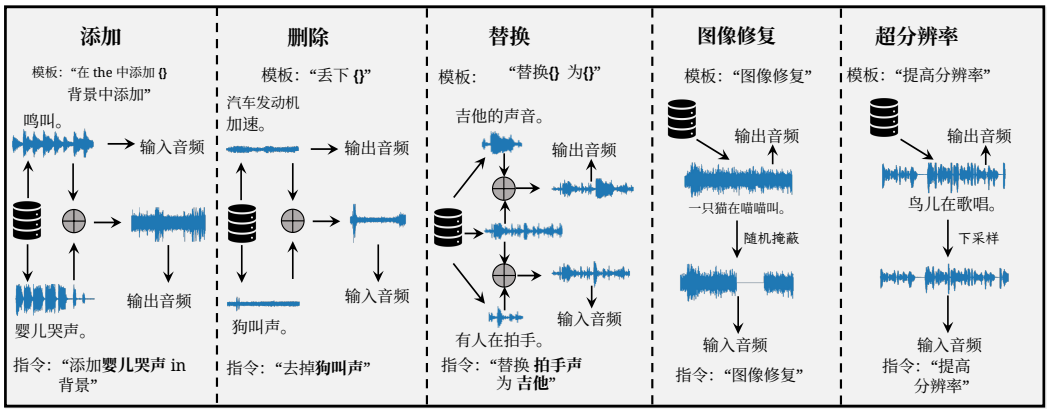


图 3：不同音频编辑任务（添加、删除、替换、图像修复和超分辨率）的三元组训练数据生成示例。

是"婴儿哭声"，因此指令为"在背景中添加婴儿哭声"。请注意，我们针对不同的编辑任务设计了多个模板。

删除 我们从文本-音频数据集中随机选取两个音频片段 A 和 B，然后将A 和 B 组合起来，得到一个新的音频片段 C。我们将 C 用作输入音频，将 A 用作输出音频，并将 B 的描述文本（或标签）填入指令模板中，生成指令。例如，指令模板为"删除 *{}*"， B 的描述文本为"狗叫声"，因此指令为"删除狗叫声"。

替换 对于替换任务，我们需要从数据集中选取三个音频片段 A、B和 C，然后分别将 B 和 C 插入 A 中（大致位于相同区域），以获得两个新的音频片段 D 和 E。我们将 D 用作输入音频，将 E 用作输出音频，并将 B 和 C 的描述文本（或标签）填入指令模板中，生成指令。例如，指令模板为"用 *{}* 替换 *{}*"， B 的描述文本为"有人拍手"， C 的描述文本为"吉他的声音"，因此指令为"用吉他替换拍手声"。

图像修复和超分辨率 对于图像修复任务，我们选择一个音频片段作为输出音频，并随机掩蔽音频的部分内容，从而得到一个新的音频片段作为输入音频。我们可以直接使用 "图像修复" 或 "图像修复音频" 这样的指令，也可以使用"图像修复：一只猫在喵喵叫"这样的指令。对于超分辨率任务，我们使用降采样后的音频作为输入音频。指令可以是"提高分辨率"或 "提高分辨率：一只鸟在歌唱"。

3.3 AUDIT 的优势

我们从三个关键方面分析了我们的方法相较于以往工作的优势。1) 我们生成了三元组数据（指令、输入数据、输出数据），用于训练文本引导的音频编辑模型，而不是执行零样本编辑，从而确保良好的编辑质量。2) 我们直接将输入音频用作扩散模型监督训练的条件，使模型能够自动学习在编辑前后保留无需编辑的部分。具体而言，我们沿通道维度拼接输入音频的潜在表示 $\boldsymbol{z}_{in}$ 和 $\boldsymbol{z}_t$，并将其输入潜在扩散模型，使模型在训练和推理期间都能"看到" $\boldsymbol{z}_{in}$ （而非其含噪版本）。3) 我们使用人类指令作为文本输入，而非对输出音频进行完整描述，这更符合实际应用场景。

4 实验设置

4.1 数据集

我们使用的数据集包括 AudioCaps [22], AudioSet [12], FSD50K [11], 和 ESC50[39]。AudioSet 是规模最大的一组音频-标签对数据集；但是，AudioSet 中的每个音频片段只有对应的标签，没有描述文本。由于 AudioSet 中包含大量人类语音，且一些音频片段与其标签的关联较弱，我们使用了 AudioSet 的一个子集 AudioSet96，其中约有 14.9 万组、涵盖 96 个类别的数据。AudioCaps 是一个包含约 4.4 万组音频-描述对的数据集，每个音频片段都对应一条包含丰富语义信息的描述文本。AudioSet 和 AudioCaps 中每个音频片段的时长约为 10 秒。FSD50K 包含约 4 万个音频片段和 200 组音频-标签对，片段时长不一，范围为 0.3 至 30 秒。我们将 FSD50K训练集划分为两个数据集：FSD50K-L（1.9 万个）和 FSD50K-S（2.2 万个），分别包含时长小于和大于 5 秒的音频片段。ESC50 是一个较小的数据集，包含 50 个类别，每个类别有四个时长约为 5 秒的音频片段，总计 2000 个音频片段。

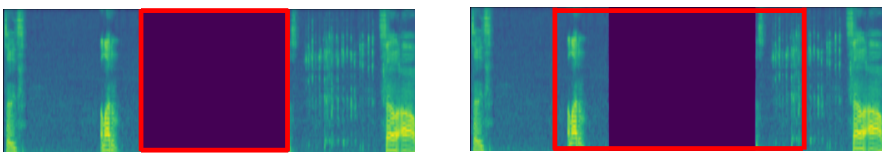
我们使用 AudioCaps、AudioSet96、FSD50K-S 和 ESC50，为五种音频编辑任务生成三元组训练数据。我们使用总计约 0.6M 条三元组数据来训练音频编辑模型。有关数据集与数据处理的更多详情，请参见附录 A。

4.2 基线系统

文本到音频生成 由于我们将基于生成模型的音频编辑基线方法作为比较对象，因此使用 AudioCaps、AudioSet96、FSD50K 和 ESC50 训练文本到音频潜在扩散模型，并使用 AudioCaps 测试集评估文本到音频模型。为展示我们生成模型的性能，我们将其与一些最先进的文本到音频生成模型进行比较 [49, 26, 27, 18]。

添加、删除和替换 我们采用 SDEdit[33] 等方法作为基线。SDEdit 使用预训练的文本到图像扩散模型，根据编辑目标描述对输入图像添加噪声，然后进行去噪。在我们的工作中，我们直接在添加任务、删除任务和替换任务中使用 SDEdit。我们使用自行训练的文本到音频潜在扩散模型。我们测试了不同的总去噪步数 N，分别为 $N = 3/4T$、$N = 1/2T$ 和 $N = 1/4T$，其中 T 是前向扩散过程中的总步数。这三种基线分别称为：1）*SDEdit-3/4T*；2）*SDEdit-1/2T*；3）*SDEdit-1/4T*。

图像修复 对于图像修复任务，我们设计了四种源自 SDEdit、但彼此略有不同的基线方法。1) *SDEdit*。我们直接使用 SDEdit：首先向输入音频添加噪声，然后以输出音频的描述文本（caption 或标签）作为文本引导，对音频进行去噪。2) *SDEdit-Rough* 和 3) *SDEdit-Precise*。在图像修复任务中，只需编辑输入音频的一部分（即被掩码的部分）。我们将无需编辑的部分称为“可观测”部分，将掩码部分称为“不可观测”部分。在去噪过程的每一步中，我们都可以在潜在空间中用真实值替换“可观测”部分。SDEdit-Rough 和 SDEdit-Precise 的区别在于，前者的“不可观测”部分是一个粗略区域，而后者的“不可观测”部分是一个精确区域。4) *SDEdit-wo-Text*。SDEdit-wo-Text 与 SDEdit-Precise 类似，但没有文本输入。



SDEdit-Precise 中的“不可观测”部分。　　SDEdit-Rough 中的“不可观测”部分。

图 4：SDEdit-Rough 与 SDEdit-Precise 之间的差异。

超分辨率 超分辨率可以视为频域中的图像修复。我们使用了三种与图像修复任务类似的基线方法：1)*SDEdit*；2) *SDEdit-Precise*；3) *SDEdit-wo-Text*。

4.3 评估指标

客观指标 我们使用对数谱距离（LSD）、弗雷歇距离（FD）和库尔贝克–莱布勒散度（KL）评估我们的文本引导的音频编辑模型，并按照 [49, 26, 27, 18] 使用 Inception分数（IS）、FD 和 KL 评估我们的文本到音频生成模型。LSD 衡量输出样本与目标样本的频谱图之间的距离。FD 衡量生成样本与目标样本之间的保真度。IS 衡量生成样本的质量和多样性。KL 衡量输出样本与目标样本之间的相关性。FD、IS 和 KL 基于最先进的音频分类模型 PANNs [25]。为实现更公平的比较，我们使用 [27] 提供的评估流程 [2] 进行客观评估。

主观指标 我们使用整体质量（Quality）衡量输出音频相较于输入音频的声音质量和自然度，并使用与编辑指令的相关性（Relevance）衡量输出音频与输入人工指令的匹配程度。每个样本将根据 Quality 和 Relevance 获得 1 到 100 分。有关评估的更多详情，请参阅附录 E。

4.4 模型配置

我们的自编码器将大小为 $1 \times H \times W$ 的梅尔频谱图压缩为大小为 $4 \times \frac{H}{4} \times \frac{W}{4}$ 的潜在表示。我们的模型使用 8 块 NVIDIA V100 GPU 训练 500K 步，每块设备的批量大小为 2。我们使用预训练文本转音频模型的权重来初始化音频编辑模型。我们的 HiFi-GAN 声码器使用 AudioSet96 和 AudioCaps 数据集，在 8 块 NVIDIA V100 GPU 上训练 200 个 epoch。关于模型配置的更多详情见附录 C。

5 结果

5.1 客观评估

添加 表 1 展示了添加任务的客观评估结果。我们的方法表现最佳，LSD 为 1.35，KL 散度为 0.92，FD（弗雷歇距离）为 21.80。与表现最好的基线方法相比，我们的方法将 FD 降低了 6.45，KL 散度降低了 0.38。

表 1：添加任务的客观评估结果。

模型	Text	LSD(↓)	KL散度(↓)	FD（弗雷歇距离）(↓)
SDEdit-3/4T	描述文本	1.54	1.68	28.87
SDEdit-1/2T	描述文本	1.43	1.38	28.75
SDEdit-1/4T	描述文本	1.38	1.30	28.25
AUDIT	指令	**1.35**	**0.92**	**21.80**

删除 表 2 展示了删除任务的客观评估结果。我们的方法表现最佳，LSD 为 1.37，KL散度为 0.95，FD（弗雷歇距离）为 22.40。与表现最佳的基线方法相比，我们的方法将 FD（弗雷歇距离）降低了 5.79，KL散度降低了 0.10。

替换 表 3 展示了替换任务的客观评估结果。我们的方法表现最佳，LSD 为 1.37，KL散度为 0.84，FD（弗雷歇距离）为 21.65。与表现最佳的基线方法相比，我们的方法将 FD（弗雷歇距离）降低了 5.07，KL散度降低了 0.31。

图像修复表4展示了修复任务的客观评估结果。我们的方法表现最佳，LSD为1.32，KL散度为0.75，FD（弗雷歇距离）为18.17。我们发现，对于基线方法，不提供作为引导的文本输入（音频的描述文本）会导致性能大幅

[2] https://github.com/haoheliu/audioldm_eval

表 2：删除任务的客观评估结果。

模型	Text	LSD(↓)	KL(↓)	FD(↓)
SDEdit-3/4T	描述文本	1.54	1.14	29.66
SDEdit-1/2T	描述文本	1.43	1.05	28.19
SDEdit-1/4T	描述文本	1.40	1.30	31.31
AUDIT	指令	**1.37**	**0.95**	**22.40**

表 3：替换任务的客观评估结果。

模型	Text	LSD(↓)	KL散度(↓)	FD（弗雷歇距离）(↓)
SDEdit-3/4T	描述文本	1.63	1.58	28.78
SDEdit-1/2T	描述文本	1.52	1.27	27.71
SDEdit-1/4T	描述文本	1.46	1.15	26.72
AUDIT	指令	**1.37**	**0.84**	**21.65**

下降；在基线方法中，SDEdit-wo-Text的KL散度和FD（弗雷歇距离）表现最差。然而，仅使用“修复”之类指令的AUDIT-wo-Text，其表现接近使用“修复 + 描述文本”之类指令的AUDIT。

表4：修复任务的客观评估结果。

模型	Text	LSD(↓)	KL散度(↓)	FD（弗雷歇距离）(↓)
SDEdit	描述文本	2.91	1.47	25.42
SDEdit-Rough	描述文本	1.64	0.98	21.99
SDEdit-Precise	描述文本	1.54	0.94	21.07
SDEdit-wo-文本	-	1.55	1.63	27.63
AUDIT-wo-文本	指令	1.37	0.81	19.03
AUDIT	指令 + 描述文本	**1.32**	**0.75**	**18.17**

超分辨率表5展示了超分辨率任务的客观评估结果。我们的方法表现最佳，LSD为1.48，KL散度为0.73，FD（弗雷歇距离）为18.14。与修复任务类似，在KL散度和FD（弗雷歇距离）方面，SDEdit-wo-Text在基线方法中表现最差。仅使用“提高分辨率”这类简单指令，我们的方法就能取得显著优于基线方法的结果，这表明我们的方法能够从低频信息中学习到充分的语义信息。

表5：超分辨率任务的客观评估结果。

模型	Text	LSD(↓)	KL(↓)	FD(↓)
SDEdit	描述文本	3.14	1.50	25.31
SDEdit-Precise	描述文本	1.75	1.17	27.81
SDEdit-wo-Text	-	1.78	1.56	32.66
AUDIT-wo-Text	指令	1.53	0.92	21.97
AUDIT	指令 + 描述文本	**1.48**	**0.73**	**18.14**

我们发现，对于图像修复和超分辨率任务，字幕作为引导的存在会显著影响基线方法的结果（在图像修复任务和超分辨率任务中，SDEdit-wo-Text 的表现最差，FD 分别为 27.63 和 32.66）。对于我们的方法，字幕作为引导的存在不会显著影响客观指标。可能的原因是，基线方法使用预训练文本转音频模型，没有文本引导时可能无法取得更好的生成效果。然而，我们的模型在大量训练数据中无法看到音频的描述文本，因此能够更好地利用上下文完成图像修复和超分辨率。

文本到音频生成 我们还在表 6 中展示了我们的文本到音频潜在扩散模型与其他文本到音频模型的对比结果。我们的模型表现最佳，FD 为 20.19、KL 散度为 1.32、IS 为 9.23，优于 AudioLDM-L-Full（FD 为 23.31、KL 散度为 1.59、IS 为 8.13）。这表明，我们的生成模型可以作为基于生成的编辑方法的强基线模型。

Tabl图 6：我们的文本到音频生成模型与其他基线模型的比较
在 AudioCaps 测试集上。

模型	数据集	FD（弗雷歇距离）(↓)	IS（初始分数）(↑)	KL散度(↓)
DiffSound [49]	AudioSet+AudioCaps	47.68	4.01	2.52
AudioGen [26]	AudioSet+AudioCaps	-	-	2.09
Make-an-Audio[18]	AudioSet+AudioCaps+13 其他	-	-	2.79
AudioLDM-L [27]	AudioCaps	27.12	7.51	1.86
AudioLDM-L-Full [27]	AudioSet+AudioCaps+2 其他	23.31	8.13	1.59
AUDIT	AudioSet96+AudioCaps + 2 其他	**20.19**	**9.23**	**1.32**

5.2 主观评估

主观评估结果如表7所示。我们报告基线中表现最佳的结果。在所有五项任务中，无论是质量还是相关性，我们的方法都明显优于基线方法。在删除任务中，我们的方法得分最高，质量为78.1，相关性为81.0。我们发现，与基线方法相比，我们的方法在添加、删除和替换任务上的提升更为显著。可能的原因是，与图像修复和超分辨率任务不同，后两者对编辑区域（掩码部分和高频区域）有明确定位，而添加、删除和替换任务需要先定位待编辑区域，同时还要确保其他区域不被修改，这对基线方法来说难度更大。我们的模型通过监督学习习得了这种能力。

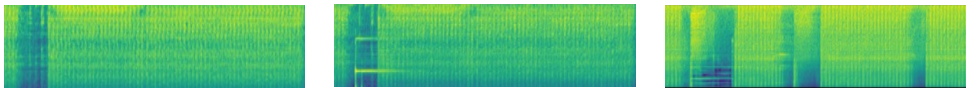
表7：主观评估。对于每项音频编辑任务，我们报告表现最佳的基线方法的结果以及我们方法的结果。

方法	Task	质量(↑)	相关性(↑)
基线	添加	60.2	56.7
AUDIT		**75.5**	**77.3**
基线	删除	54.4	48.2
AUDIT		**78.1**	**81.0**
基线	替换	57.7	47.6
AUDIT		**72.5**	**74.5**
基线	图像修复	65.8	66.3
AUDIT		**75.2**	**78.7**
基线	超分辨率	61.8	59.9
AUDIT		**74.6**	**76.3**

5.3 案例研究

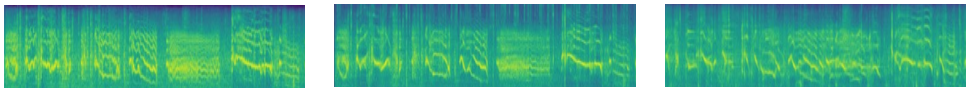
我们还展示了一些案例研究。图5展示了一个添加任务的案例。输入音频的描述文本是 “机关枪的声音”，编辑目标是在开头添加铃声。AUDIT 能够在正确的区域准确地进行音频编辑，同时不修改无需编辑的音频片段。图6展示了一个删除任务的案例。输入音频的描述文本是 “一只鸟持续鸣叫，同时背景中有鸭子在水里嘎嘎叫”，编辑目标是删除鸭叫声。我们的方法成功移除了背景声，并保留了鸟鸣声，但是鸟鸣声

在 SDEdit 编辑结果中被错误地修改了。这表明我们的方法能够更好地确保无需编辑的音频片段不被修改。图7展示了一个修复任务的案例。输入音频的描述文本是 “有人正在敲打电脑”。虽然 AUDIT 和基线方法生成的结果在语义上都正确，但 AUDIT 生成的结果更加自然，也更符合上下文。



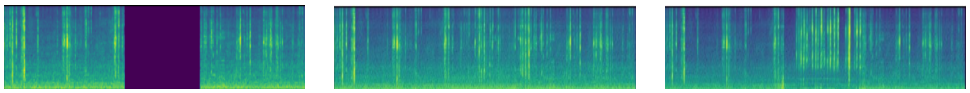
输入　　通过 AUDIT 生成　　通过 SDEdit 生成

图5：添加任务的一个案例。输入音频的描述文本为“机关枪的声音”，编辑目标是在开头添加铃声。



输入　　通过 AUDIT 生成　　通过 SDEdit 生成

图6：删除任务的一个案例。输入音频的描述文本为“一只鸟持续鸣叫，背景中有一只鸭子在水中呱呱叫”，编辑目标是删除鸭叫声。



输入　　由 AUDIT 生成　　由 SDEdit-Precise 生成

图7：图像修复任务的一个案例。输入音频的描述文本为“有人正在敲击电脑键盘”。

6 结论

在本研究中，我们提出了一种名为 AUDIT 的音频编辑模型，可根据人工文本指令执行不同的编辑任务（例如添加、删除、替换、图像修复和超分辨率）。具体而言，我们使用生成的三元组训练数据（指令、输入音频、输出音频）训练了一个文本引导的潜在扩散模型。该模型只需简单的人工指令作为引导，无需提供输出音频的描述；同时，它能够准确地执行音频编辑，而不修改无需编辑的音频片段。AUDIT 在五种不同的音频编辑任务上，客观和主观指标均达到了最先进的性能。未来，我们将利用该框架探索更多音频编辑任务，并实现对音频编辑更精确的控制。

本研究符合微软的负责任人工智能原则。

参考文献

[1] Sherif Abdulatif、Ruizhe Cao 和 Bin Yang。Cmgan：用于单声道语音增强的基于 Conformer 的度量 GAN。*arXiv* 预印本 *arXiv:2209.11112*，2022年。[2] Masanobu Abe、Satoshi Nakamura、Kiyohiro Shikano 和 Hisao Kuwabara。通过向量量化进行语音转换。日本声学学会志（*E*），11(2):71–76，1990年。[3] Andrea Agostinelli、Timo I Denk、Zalán Borsos、Jesse Engel、Mauro Verzetti、Antoine Caillon、Qingqing Huang、Aren Jansen、Adam Roberts、Marco Tagliasacchi 等。Musiclm：从文本生成音乐。*arXiv* 预印本 *arXiv:2301.11325*，2023年。[4] Omri Avrahami、Dani Lischinski 和 Ohad Fried。用于自然图像文本驱动编辑的混合扩散。载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，第18208–18218页，2022年。[5] Zalán Borsos、Matt Sharifi 和 Marco Tagliasacchi。Speechpainter：文本条件语音修复。*arXiv* 预印本 *arXiv:2202.07273*，2022年。[6] Tim Brooks、Aleksander Holynski 和 Alexei A Efros。Instructpix2pix：学习遵循图像编辑指令。*arXiv* 预印本 *arXiv:2211.09800*，2022年。[7] Xin Chen、Wei Chu、Jinxi Guo 和 Ning Xu。使用非平行数据进行歌声转换。载于*2019*年 *IEEE* 多媒体信息处理与检索会议（*MIPR*），第292–296页。IEEE，2019。[8] Jooyoung Choi、Sungwon Kim、Yonghyun Jeong、Youngjune Gwon 和 Sungroh Yoon。Ilvr：用于去噪扩散概率模型的条件化方法。*arXiv* 预印本 *arXiv:2108.02938*，2021年。[9] Ondˇrej Cífka、Umut ¸Sim¸sekli 和 Gaël Richard。Groove2groove：利用合成数据监督的单样本音乐风格迁移。*I EEE/ACM* 音频、语音与语言处理汇刊，28:2638–2650，2020年。[10] Pirmin P Ebner 和 Amr Eltelt。使用生成对抗网络进行音频修复。*arXiv* 预印本 *arXiv:2003.07704*，2020 年。[11] Eduardo Fonseca、Xavier Favory、Jordi Pons、Frederic Font 和 Xavier Serra。Fsd50k：人工标注声音事件的开放数据集。*IEEE/ACM* 音频、语音与语言处理汇刊，30:829–852，2021年。[12] Jort F Gemmeke、Daniel PW Ellis、Dylan Freedman、Aren Jansen、Wade Lawrence、R Channing Moore、Manoj Plakal 和 Marvin Ritter。Audio set：用于音频事件的本体与人工标注数据集。载于*2017*年 *IEEE* 国际声学、语音与信号处理会议（*ICASSP*），第776–780页。IEEE，2017年。[13] Shuyang Gu、Dong Chen、Jianmin Bao、Fang Wen、Bo Zhang、Dongdong Chen、Lu Yuan 和 Baining Guo。用于文本到图像合成的向量量化扩散模型。载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，第10696–10706页，2022年。[14] Amir Hertz、Ron Mokady、Jay Tenenbaum、Kfir Aberman、Yael Pritch 和 Daniel Cohen-Or。通过交叉注意力控制实现提示词到提示词图像编辑。*arXiv* 预印本 *arXiv:2208.01626*，2022年。[15]J onathan Ho 和 Tim Salimans。无分类器扩散引导。*arXiv* 预印本 *arXiv:2207.12598*，2022年。[16] Jonathan Ho、Ajay Jain 和 Pieter Abbeel。去噪扩散概率模型。神经信息处理系统进展，33:6840–6851，2020年。[17] Qingqing Huang、Daniel S Park、Tao Wang、Timo I Denk、Andy Ly、Nanxin Chen、Zhengdong Zhang、Zhishuai Zhang、Jiahui Yu、Christian Frank 等。Noise2music：使用扩散模型进行文本条件音乐生成。*arXiv* 预印本 *arXiv:2302.03917*，2023年。

[18] Rongjie Huang、Jiawei Huang、Dongchao Yang、Yi Ren、Luping Liu、Mingze Li、Zhenhui Ye、Jinglin Liu、Xiang Yin 和 Zhou Zhao。Make-an-audio：通过提示增强扩散模型实现文本到音频生成。*arXiv* 预印本 *arXiv:2301.12661*，2023年。[19] Phillip Isola、Jun-Yan Zhu、Tinghui Zhou 和 Alexei A Efros。利用条件对抗网络进行图像到图像转换。载于*IEEE* 计算机视觉与模式识别会议论文集，第1125–1134页，2017年。[20] Nal Kalchbrenner、Erich Elsen、Karen Simonyan、Seb Noury、Norman Casagrande、Edward Lockhart、Florian Stimberg、Aaron Oord、Sander Dieleman 和 Koray Kavukcuoglu。高效的神经音频合成。载于国际机器学习会议，第2410–2419页。PMLR，2018年。[21] Mihir Kale 和 Abhinav Rastogi。面向数据到文本任务的文本到文本预训练。*arXiv* 预印本 *arXiv:2005.10433*，2020年。[22] Chris Dongjoo Kim、Byeongchang Kim、Hyunmin Lee 和 Gunhee Kim。AudioCaps：为真实场景中的音频生成字幕。载于计算语言学协会北美分会*2019*年会议：人类语言技术论文集，第*1*卷（长文与短文），第119–132页，2019年。[23] Diederik P Kingma 和 Max Welling。自编码变分贝叶斯。*arXiv* 预印本 *arXiv:1312.6114*，2013年。[24] Jungil Kong、Jaehyeon Kim 和 Jaekyoung Bae。HiFi-GAN：用于高效、高保真语音合成的生成对抗网络。神经信息处理系统进展，33：17022–17033，2020年。[25] Qiuqiang Kong、Yin Cao、Turab Iqbal、Yuxuan Wang、Wenwu Wang 和 Mark D Plumbley。PANNs：用于音频模式识别的大规模预训练音频神经网络。*IEEE/ACM* 音频、语音与语言处理汇刊，28：2880–2894，2020年。[26] Felix Kreuk、Gabriel Synnaeve、Adam Polyak、Uriel Singer、Alexandre Défossez、Jade Copet、Devi Parikh、Yaniv Taigman 和 Yossi Adi。AudioGen：文本引导的音频生成。*arXiv* 预印本 *arXiv:2209.15352*，2022年。[27] Haohe Liu、Zehua Chen、Yi Yuan、Xinhao Mei、Xubo Liu、Danilo Mandic、Wenwu Wang 和 Mark D Plumbley。AudioLDM：利用潜在扩散模型进行文本到音频生成。*arXiv* 预印本 *arXiv:2301.12503*，2023年。[28] Songxiang Liu、Yuewen Cao、Dan Su 和 Helen Meng。DiffSVC：用于歌声转换的扩散概率模型。载于*2021*年 *IEEE* 自动语音识别与理解研讨会（*ASRU*），第741–748页。IEEE，2021年。[29] Ilya Loshchilov 和 Frank Hutter。解耦权重衰减正则化。*arXiv* 预印本 *arXiv:1711.05101*，2017年。[30] Chien-Yu Lu、Min-Xin Xue、Chia-Che Chang、Che-Rung Lee 和 Li Su。随心演奏：音色增强的多模态音乐风格迁移。载于*AAAI* 人工智能会议论文集，第33卷，第1061–1068页，2019年。[31] Andreas Lugmayr、Martin Danelljan、Andres Romero、Fisher Yu、Radu Timofte 和 Luc Van Gool。RePaint：使用去噪扩散概率模型进行图像修复。载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，第11461–11471页，2022年。[32] Andrés Marafioti、Piotr Majdak、Nicki Holighaus 和 Nathanaël Perraudin。GACELA：用于音乐长音频修复的生成式对抗上下文编码器。*IEEE* 信号处理专题期刊，15(1)：120–131，2021年。doi：10.1109/JSTSP.2020.3037506。[33] Chenlin Meng、Yang Song、Jiaming Song、Jiajun Wu、Jun-Yan Zhu 和 Stefano Ermon。SDEdit：利用随机微分方程进行图像合成与编辑。*arXiv* 预印本 *arXiv:2108.01073*，2021年。

[34] Seyed Hamidreza Mohammadi 和 Alexander Kain。语音转换系统概述。*Speech Communication*，88：65–82，2017年。[35] Ondˇrej Mokr`y 和 Pavel Rajmic。音频修复：重新审视与重新加权。*IEEE/ACM* 音频、语音与语言处理汇刊，28：2906–2918，2020年。[36] Chong Mou、Xintao Wang、Liangbin Xie、Jian Zhang、Zhongang Qi、Ying Shan 和 Xiaohu Qie。T2I-Adapter：通过学习适配器挖掘文本到图像扩散模型的更多可控能力。*arXiv* 预印本 *arXiv:2302.08453*，2023年。[37] Aaron van den Oord、Sander Dieleman、Heiga Zen、Karen Simonyan、Oriol Vinyals、Alex Graves、Nal Kalchbrenner、Andrew Senior 和 Koray Kavukcuoglu。Wavenet：一种原始音频生成模型。*arXiv* 预印本 *arXiv:1609.03499*，2016年。[38] Marco Pasini。Melgan-vc：利用频谱图对任意长度样本进行语音转换和音频风格迁移。*arXiv* 预印本 *arXiv:1910.03713*，2019年。[39] Karol J Piczak。ESC：环境声音分类数据集。载于第 *23*届 *ACM* 多媒体国际会议论文集，第1015–1018页，2015年。[40] Wei Ping、Kainan Peng、Kexin Zhao 和 Zhao Song。Waveflow：一种用于原始音频的紧凑型基于流的模型。载于国际机器学习会议，第7706–7716页。PMLR，2020年。[41] Adam Polyak、Lior Wolf、Yossi Adi 和 Yaniv Taigman。无监督跨域歌声转换。*arXiv* 预印本 *arXiv:2008.02830*，2020年。[42] Vadim Popov、Ivan Vovk、Vladimir Gogoryan、Tasnima Sadekova、Mikhail Kudinov 和 Jiansheng Wei。基于扩散的语音转换及快速最大似然采样方案。*arXiv* 预印本 *arXiv:2109.13821*，2021年。[43] Robin Rombach、Andreas Blattmann、Dominik Lorenz、Patrick Esser 和 Björn Ommer。利用潜在扩散模型实现高分辨率图像合成。载于*IEEE/CVF* 计算机视觉与模式识别会议论文集，第10684–10695页，2022年。[44] Olaf Ronneberger、Philipp Fischer 和 Thomas Brox。U-Net：用于生物医学图像分割的卷积网络。载于医学图像计算与计算机辅助干预——*MICCAI 2015*：第*18*届国际会议，德国慕尼黑，*2015*年*10*月*5*日至*9*日，论文集，第*III*部分*18*，第234–241页。Springer，2015年。[45] Chitwan Saharia、William Chan、Huiwen Chang、Chris Lee、Jonathan Ho、Tim Salimans、David Fleet 和 Mohammad Norouzi。Palette：图像到图像扩散模型。载于*ACM SIGGRAPH 2022* 会议论文集，第1–10页，2022年。[46] Yang Song、Jascha Sohl-Dickstein、Diederik P Kingma、Abhishek Kumar、Stefano Ermon 和 Ben Poole。通过随机微分方程进行基于分数的生成建模。*arXiv* 预印本 *arXiv:2011.13456*，2020年。[47] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Łukasz Kaiser 和 Illia Polosukhin。Attention is all you need。神经信息处理系统进展，30，2017年。[48] Shih-Lun Wu 和 Yi-Hsuan Yang。Musemorphose：使用单个 Transformer 变分自编码器实现全曲及细粒度音乐风格迁移。*arXiv* 预印本 *arXiv:2105.04090*，2021年。[49] Dongchao Yang、Jianwei Yu、Helin Wang、Wen Wang、Chao Weng、Yuexian Zou 和 Dong Yu。DiffSound：用于文本到声音生成的离散扩散模型。*arXiv* 预印本 *arXiv:2207.09983*，2022年。[50] Neil Zeghidour、Alejandro Luebs、Ahmed Omran、Jan Skoglund 和 Marco Tagliasacchi。Soundstream：端到端神经音频编解码器。*IEEE/ACM* 音频、语音与语言处理汇刊，30：495–507，2021年。[51] Lvmin Zhang 和 Maneesh Agrawala。为文本到图像扩散模型添加条件控制。*arXiv* 预印本 *arXiv:2302.05543*，2023年。

[52] Hang Zhou、Ziwei Liu、Xudong Xu、Ping Luo 和 Xiaogang Wang。视觉增强的深度音频修复。发表于*IEEE/CVF*国际计算机视觉会议论文集，第 283–292 页，2019。

A 数据集与数据处理

表 8 展示了我们使用和生成的数据集的一些详情。所有音频数据的采样率均为 16KHz。对于 AudioSet、AudioCaps 和 AudioSet96，我们将所有音频补齐或截断至 10 秒。对于 FSD50K，我们将短于 5 秒的音频补齐至 5 秒，并将长于 5 秒的音频截断或补齐至 10 秒。所有生成的音频数据长度均为 10 秒。我们将 10 秒音频转换为尺寸为 80×624 的梅尔频谱图，使用 256 的帧移、1024 的窗口大小以及大小为 80 的梅尔频带，覆盖 0 至 8000 Hz 的频率范围。自编码器将梅尔频谱图压缩为尺寸为 $4 \times 10 \times 78$ 的潜在表示。

表 8：我们使用的音频文本数据集及生成的音频编辑数据集详情

数据集	小时	数量	Text
AudioSet	5800	2M	标签
AudioSet96	414	149K	标签
AudioCaps	122	44K	描述文本
FSD50K	108	51K	标签
FSD50K-S	31	22K	标签
FSD50K-L	53	19K	标签
ESC50	3	2K	标签
Task	**数据集**	**数量**	**Text**
生成	AudioCaps、AudioSet96、FSD50K、ESC50	243K	标签或描述
添加	AudioCaps、AudioSet96、FSD50K-S、ESC50	71K	指令
删除	AudioCaps、AudioSet96、FSD50K-S、ESC50	71K	指令
替换	AudioSet96、FSD50K-S、ESC50	50K	指令
图像修复	AudioSet96、AudioCaps	193K	指令
超分辨率	AudioSet96、AudioCaps	193K	指令

B 文本指令模板

一些指令模板示例：

“在开头添加 {} ” “在开头添加 {} ” “在结尾添加 {} ” “在中间添加 {} ” “在背景中添加 {} ” “删除 {}” “移除 {}” “将 {} 替换为 {}” “用 {}替换 {} ” “图像修复” “图像修复” “修复图像 {}” “图像修复： {}” “提高分辨率” “提高分辨率： {}” “执行超分辨率处理” “执行超分辨率处理： {}”

C模型详细信息

表9展示了有关我们的音频编辑和音频生成模型的更多细节。我们使用批量大小为32（每个设备8）的配置，在8块NVIDIA V100 GPU上训练自编码器模型，共进行50000步，学习率为 $7.5e-5$。对于音频编辑和U-Net音频生成扩散，我们使用批量大小为8的配置，在8块NVIDIA V100 GPU上训练，共进行500000步，学习率为 $5e-5$。自编码器和扩散模型均使用AdamW[29] 作为优化器，参数为（β_1，β_2）=（0。9，0。999），权重衰减为 $1e-2$。我们遵循官方代码仓库的做法训练HiFi-GAN声码器。

表9：我们的音频编辑和音频生成模型的详细信息

模型	配置	
自编码器	参数数量	83M
	输入/输出通道	1
	潜在通道	4
	下采样/上采样块数量	4
	块输出通道数	(128, 256, 512, 512)
	激活函数	SiLU
T5文本编码器	参数数量	109M
	输出通道	768
	最大长度	300
编辑扩散U-Net	参数数量	859M
	输入通道	8
	输出通道	4
	下采样/上采样块数量	4
	块输出通道数	(320, 640, 1280, 1280)
	注意力头	8
	交叉注意力维度	768
	激活函数	SiLU
生成式扩散 U-Net	参数数量	859M
	输入通道	4
	输出通道	4
	下采样/上采样块数量	4
	块输出通道数	(320, 640, 1280, 1280)
	注意力头	8
	交叉注意力维度	768
	激活函数	SiLU
HiFi-GAN	采样率	16000
	梅尔频带数	80
	帧移	256
	窗口大小	1024

D 无分类器引导

[15] 提出使用无分类器引导来权衡多样性和样本质量。无分类器引导策略已广泛应用于条件扩散模型中。我们的模型还有两个额外条件，c_{text} 和 $\boldsymbol{z}_{in}$。然而，在训练过程中，我们只考虑帮助模型学习 $p(\boldsymbol{z}_t|\boldsymbol{z}_{in})$ 的边缘分布（而不显式学习 $p(\boldsymbol{z}_t|c_{text})$）。因此，在训练过程中，我们以一定概率将文本遮蔽（将其替换为空文本 $\emptyset$），以学习无文本条件得分 $\boldsymbol{\epsilon_\theta}(\boldsymbol{z}_t,\ t,\ \boldsymbol{z}_{in})$。然后，根据贝叶斯公式 $p(\boldsymbol{z}_t|c_{text}) \propto p(\boldsymbol{z}_t|c_{text},\ \boldsymbol{z}_{in})\, p(\boldsymbol{z}_{in}|\boldsymbol{z}_t,\ c_{text})$，我们可以推导出公式 1 中的得分关系，这对应于无分类器引导。

自由引导公式 2。这里，参数 $s \geq 1$ 是用于平衡样本多样性和质量的引导系数。

$$\nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_{in}|\boldsymbol{z}_t, c_{text}) = \nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_t|c_{text}, \boldsymbol{z}_{in}) - \nabla_{\boldsymbol{z}_t} \log p(\boldsymbol{z}_t|\boldsymbol{z}_{in}) \tag{1}$$

$$\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t, \boldsymbol{z}_{in}, c_{text}) = \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t, \boldsymbol{z}_{in}, \emptyset) + s \cdot (\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t, \boldsymbol{z}_{in}, c_{text}) - \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\boldsymbol{z}_t, t, \boldsymbol{z}_{in}, \emptyset)) \tag{2}$$

E 评估

对于每个音频编辑任务，我们的人工评估集包含从测试集中随机选取的十个样本。十位评分者根据质量和相关性这两项评价指标，采用 1 到 100 分制对每个样本进行评分。