arXiv:2301.12662v1 [cs.SD] 30 Jan 2023

SingSong：从歌唱生成音乐伴奏

Chris Donahue * **Antoine Caillon** * [1] **Adam Roberts** [*] **Ethan Manilow Philippe Esling** [1]
Andrea Agostinelli Mauro Verzetti Ian Simon Olivier Pietquin Neil Zeghidour Jesse Engel

Google Research

摘要

我们提出了 SingSong，这是一个生成器乐音乐以伴奏输入人声的系统，有望为音乐家和非音乐家提供一种直观的新方式，让他们用自己的声音创作音乐。为实现这一目标，我们借鉴了音乐源分离和音频生成领域的最新进展。具体而言，我们将最先进的源分离算法应用于大规模音乐音频语料库，以生成由人声和器乐源组成的对齐样本对。随后，我们对 AudioLM（Borsos 等人，2022年）这一用于无条件音频生成的先进方法进行改造，使其适用于条件式“音频到音频”生成任务，并使用源分离得到的（人声、器乐）样本对对其进行训练。在使用相同输入人声的成对比较中，听众明显更偏好由 SingSong 生成的器乐，相较于强大的检索基线生成的器乐。

1. 引言

歌唱是参与音乐活动最直观的方式之一，音乐家和非音乐家都可以轻松参与。虽然跟着现有音乐一起唱很常见，但歌唱也可能成为控制音乐生成系统的一种直观方式，让任何会唱歌的人都能以轻松、有参与感的方式创作音乐。

在这项工作中，我们推出了SingSong （图1），这是一个能够生成伴奏音乐音频、与输入人声音频步调一致的系统；也就是说，输出伴奏可以直接与输入人声混合，创作出以输入人声为特色、和谐连贯的音乐。SingSong 利用了音乐技术两个关键领域的进展：源分离和音频生成建模。我们采用Kim 等人（2021年）提出的现成源分离算法，将一个规模庞大且多样化的音乐语料库（1百万首曲目）分离为人声与伴奏的对齐样本对，作为本任务的平行数据。随后，我们对 AudioLM（Borsos 等人，2022年）进行改造，使其适用于给定人声来生成伴奏的条件式“音频到音频”生成建模。AudioLM 是一种无条件音频生成模型，采用中间表示层级；我们使用源分离数据以监督方式对其进行训练。

这项工作的一个关键挑战，是构建一个系统，使其能够从训练期间观察到的源分离人声输入中泛化到用户使用系统时可能提供的真实世界分离人声。初步实验得到的模型存在明显偏向：它们会根据源分离人声中几乎听不见的伪影重建伴奏；若输入分离人声，这些模型便会生成毫无意义的输出。为提升泛化能力，我们针对输入人声提出了两种特征化策略：（1）向人声输入添加噪声，以掩盖伪影；（2）仅使用 AudioLM 中最粗粒度的中间表示作为条件输入。在一项与感知相关的指标上，这两种特征化策略结合使用后，相较于默认的 AudioLM 特征化，使分离人声的表现提升了 55%。

在一项成对研究中，听众听到两段人声相同的人声-器乐混音，并对 SingSong 生成的器乐表现出显著偏好，相较之下，他们更喜欢 SingSong 的器乐，而非强大的检索基线所提供的器乐。[1] 该基线将人声的音乐特征（节拍和调性）作为查询，以检索具有相似特征的人工创作的器乐。与这一检索基线提供的器乐相比，听众有 66% 的时候更偏好 SingSong 生成的器乐。此外，即使与真实器乐伴奏相比，听众也有 34% 的时候更偏好 SingSong 生成的器乐。

[*]共同贡献 [1]IRCAM -索邦大学´（本工作是在访问 Google Research 期间完成的）。联系作者：Jesse Engel< jesseengel@google.com>。

[1]示例：https://g.co/magenta/singsong

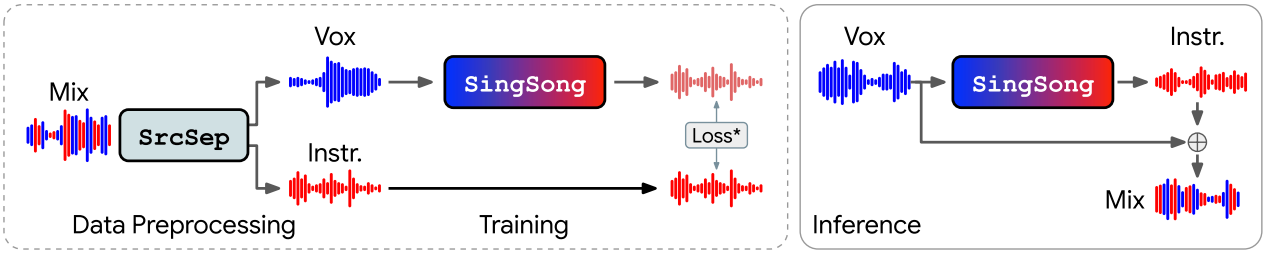


图*1.* SingSong 生成伴奏来配合输入人声，从而让用户能够创作以自己的声音为特色的音乐。（**左侧**）我们通过对大量音乐音频语料库应用现成的源分离算法，为此任务生成海量合成数据，并用这些数据训练一个根据人声生成器乐的生成模型。（**右侧**）在推理时，SingSong接收用户的人声，并输出与之配合的器乐；将其与输入简单混合，即可创作出连贯的音乐。*请注意，训练期间，我们在离散音频特征而非波形上计算损失（第3.3节）。

总而言之，我们的主要贡献如下：

- 我们率先运用生成式建模，为人声输入创作连贯的器乐伴奏。
- 我们率先提出利用源分离为音频到音频伴奏生成训练数据。
- 我们将最先进的无条件音频生成模型改造为用于条件音频到音频建模的模型。

2. 相关工作

伴奏生成。与我们工作最相似的是一款名为 Microsoft Song- smith 的商业产品，其基于 Simon 等人 (2008年)。Songsmith 从输入人声中提取音高信息，并预测适用于该旋律的一系列符号化和弦标签。随后，系统将预测出的和弦用作查询，以检索合适的、由人类创作的符号化器乐伴奏，并将其与输入人声混合。我们直接与自行构建的类似检索基线进行比较。Lattner &Grachten (2019年) 生成符号化的底鼓伴奏，而 Grachten 等人 (2020年) 生成贝斯伴奏。Wu 等人 (2022年) 根据不含鼓的音乐音频生成鼓轨音频。本文探讨的任务范围更广：仅根据人声演唱生成完整的伴奏轨（其中可能包含鼓、贝斯及其他和声元素）。此外，还有一类关于符号化和声编配的研究，其任务是针对符号化旋律输入预测合适的和弦标签（Paiement 等人，2006；Yeh 等人，2020）。若要将这类方法用于人声伴奏，还需要额外进行人声转录和和弦编配。

音乐音频生成。音乐音频生成是一项具有挑战性的任务，因为它涉及对多种不同音频尺度的建模，从局部信息（例如音色、瞬态）到全局信息（例如音乐结构、流派）。一些无条件音乐生成的研究直接对时域音乐波形进行建模（van den Oord 等人，2016；Mehri 等人，2017；Donahue 等人，2019；Goel 等人，2022）。这些方法只能对狭窄分布的音乐音频进行建模（通常是钢琴音乐）。van den Oord 等人（2017）提出了一种对音乐音频的可逆离散表示进行生成建模的方法。Dieleman 等人（2018）对钢琴音乐音频的分层离散特征进行建模；Dhariwal 等人（2020）采用了类似方法，创建了首个能够生成广泛音乐音频的模型 Jukebox。Hawthorne 等人（2022a）和 Borsos 等人（2022）也对钢琴音乐的离散音乐特征表示进行建模。大量研究聚焦于以符号表示（Hawthorne 等人，2018；Engel 等人，2019；Hawthorne 等人，2022b）或演奏特征（Caillon 和 Esling，2021；Engel 等人，2020）为条件的可控音乐音频生成。在我们的工作中，我们旨在对以人声音频为条件的广泛分布的伴奏音频进行建模。

与本研究同期，Agostinelli 等人(2023) 提出了 MusicLM，该方法改造 AudioLM，使其能够根据输入文本描述生成广泛的音乐音频。MusicLM 还能够根据用户通过歌唱或哼唱表达的旋律提示生成与之对应的音乐旋律。而我们的目标是生成与输入人声步调一致的器乐，使二者能够和谐地混合在一起——输入人声不仅像在 MusicLM 中那样用于引导音乐生成，还会直接呈现在输出音乐中。

3. 任务定义与方法

我们将人声伴奏任务定义为一个条件生成建模问题，其目标是针对人声波形 $\boldsymbol{x}$，对适当的伴奏波形 $\boldsymbol{y}$ 的分布 $P(\boldsymbol{y} \mid \boldsymbol{x})$ 进行建模，其中两种波形均为单声道，长度为 T 秒，并以某个采样率 f_s 进行采样。形式化地说，$\boldsymbol{x}$ 和 $\boldsymbol{y}$ 都是 $\mathbb{R}^{f_s T}$ 中的向量。给定

用户的人声 $\boldsymbol{x}\prime$ 后，我们可以通过采样 $\boldsymbol{y}\prime \sim P(\boldsymbol{y} \mid \boldsymbol{x} = \boldsymbol{x}\prime)$ 并对输出进行线性混合（例如 $\boldsymbol{x}\prime + \boldsymbol{y}\prime$），生成包含这些人声的音乐。

3.1 音频编码代理分布的建模

直接对波形分布进行建模在经验上颇具挑战，因为表示音频需要较高的采样率。本文采用 van den Oord 等人 van den Oord 等人（2017年）首次提出的一种做法，用于无条件音频生成建模；我们转而对离散音频编码进行建模。这种方法需要一个离散编解码器，由一对函数$\texttt{Enc}$：$\mathbb{R}^{f_s T} \to \mathbb{V}^{f_k T}$和$\texttt{Dec}$：$\mathbb{V}^{f_k T} \to \mathbb{R}^{f_s T}$构成，其中 $\mathbb{V}$ 是某个有限词汇表，f_k 是音频编码率，而$\texttt{Dec}$近似于$\texttt{Enc}^{-1}$。

利用这两个函数，我们可以转而对 Enc 生成的编码上的“代理”分布进行建模，并借助 Dec 近似波形分布。具体而言，在无条件场景中，目标是对波形 $\boldsymbol{w}$ 上的分布进行建模，我们可以改为将$P(\hat{\boldsymbol{w}})$作为代理进行建模，其中$\hat{\boldsymbol{w}} = \text{Enc}(\boldsymbol{w})$。要从该分布中采样音频，我们首先采样$\hat{\boldsymbol{w}}\prime \sim P(\hat{\boldsymbol{w}})$，然后输出 $\text{Dec}(\hat{\boldsymbol{w}}\prime)$。由于 $f_k \ll f_s$ 在实践中，对这一代理进行建模在经验上是可行的。

3.2. AudioLM 基础知识

在这项工作中，我们的方法基于 AudioLM（Borsos 等人，2022年），这是一种先进的无条件音频生成模型，它采用因式分解方法对多种音频编码的代理分布进行建模。

声学编码 AudioLM 中的离散编解码器是一个预训练的 SoundStream（Zeghidour 等人，2021年）模型，包含编码器 Enc：$\mathbb{R}^{f_s T} \to \mathbb{V}^{50T \times 12}$ 以及对应的解码器 Dec，其中 $f_s = 16\text{kHz}$ 且 $|\mathbb{V}| = 1024$。Enc 的输出称为声学编码。SoundStream 使用残差矢量量化方案，以 50Hz 的速率生成 12 维声学编码向量——前 4 维和其余 8 维分别为粗粒度声学编码和细声学编码。AudioLM 将这些粗粒度编码和细粒度编码“展平”，使每秒音频分别对应 200 个和 400 个编码。我们将波形 $\boldsymbol{w}$ 的展平粗粒度声学编码记为 $\text{Coarse}(\boldsymbol{w})$，将展平细粒度声学编码记为 $\text{Fine}(\boldsymbol{w})$，二者可直接重新组合为 $\text{Enc}(\boldsymbol{w})$。请注意，由于采用残差量化方案，SoundStream 仅凭粗粒度编码就能解码音频，但使用粗粒度和细粒度编码进行解码可获得更高的音频保真度。

语义编码 过去的音乐音频生成模型，例如 Jukebox（Wu 等人，2022年），曾使用更大的模型直接对类似于声学编码的表示进行建模。AudioLM 则提出使用更小的模型来建模声学编码与低速率 语义编码之间的联合分布，这些语义编码源自语义编码器$\texttt{Sem}$：$\mathbb{R}^{f_s T} \to \mathbb{S}^{25T}$ 其运行频率为 25Hz，并使用自监督损失进行训练，而非声学重建损失。语义编码器由经过预训练的 w2v-BERT 模型（Chung 等人，2021年）中的中间表示构成，这些表示经过 k均值聚类量化，其中$k = |\mathbb{S}| = 1024$。

AudioLM 是由三个模型级联构成的系统，可生成速率越来越高的编码。具体而言，AudioLM 将语义编码与声学编码的联合分布分解为：

$$\begin{aligned} &P(\texttt{Sem}(\boldsymbol{w}), \texttt{Enc}(\boldsymbol{w})) = \\ &\quad P(\texttt{Fine}(\boldsymbol{w}) \mid \texttt{Coarse}(\boldsymbol{w})) \\ &\quad \cdot P(\texttt{Coarse}(\boldsymbol{w}) \mid \texttt{Sem}(\boldsymbol{w})) \\ &\quad \cdot P(\texttt{Sem}(\boldsymbol{w})), \end{aligned}$$

基于以下假设：细粒度$(\boldsymbol{w})$ ⫫ 语义$(\boldsymbol{w})$ | 粗粒度$(\boldsymbol{w})$，即在给定粗粒度编码的情况下，语义编码无助于生成细声学编码。每个模型都是自回归、仅含解码器的 Transformer（Vaswani 等人，2017年）“语言模型”，分别进行训练；条件化则通过简单地串接两个序列来实现。

使用 AudioLM 生成音频时，首先依次对这三个生成模型进行采样。随后，生成的语义编码作为副产品被丢弃，而生成的声学编码则输入 Dec，以生成波形。

3.3. 调整 AudioLM 以生成伴奏

本研究并不直接对乐器波形 $\boldsymbol{y}$ 在给定人声波形 $\boldsymbol{x}$的条件下建模，而是调整 AudioLM，使其对给定人声编码时的乐器编码代理分布进行建模。具体而言，我们对$P(\text{Sem}(\boldsymbol{y}), \text{Enc}(\boldsymbol{y}) \mid \text{Feats}(\boldsymbol{x}))$建模，其中 Sem 和 Enc 来自 AudioLM，而 Feats：$\mathbb{R}^{f_s T} \to \mathbb{X}^{f_x T}$ 是一个用于从人声 $\boldsymbol{x}$中提取离散特征的函数。我们采用了一个与 AudioLM 类似的三阶段分解：

$$\begin{aligned} &P(\texttt{Sem}(\boldsymbol{y}), \texttt{Enc}(\boldsymbol{y}) \mid \texttt{Feats}(\boldsymbol{x})) = \\ &\quad P(\texttt{Fine}(\boldsymbol{y}) \mid \texttt{Coarse}(\boldsymbol{y})) \\ &\quad \cdot P(\texttt{Coarse}(\boldsymbol{y}) \mid \texttt{Sem}(\boldsymbol{y}), \texttt{Feats}(\boldsymbol{x})) \\ &\quad \cdot P(\texttt{Sem}(\boldsymbol{y}) \mid \texttt{Feats}(\boldsymbol{x})), \end{aligned}$$

基于类似的独立性假设，即仅凭粗粒度伴奏编码即可重建细粒度伴奏编码。[2]

对于第 *3* 阶段——即根据粗粒度声学编码预测细声学编码的模型——我们沿用 AudioLM 的相同方法，也就是仅含解码器的 Transformer 语言模型。为方便起见，我们将前两个阶段合并为一个

[2]形式上，$\text{Fine}(\boldsymbol{y})$ ⫫ $\text{Sem}(\boldsymbol{y})$，$\text{Feats}(\boldsymbol{x})$ | $\text{Coarse}(\boldsymbol{y})$。

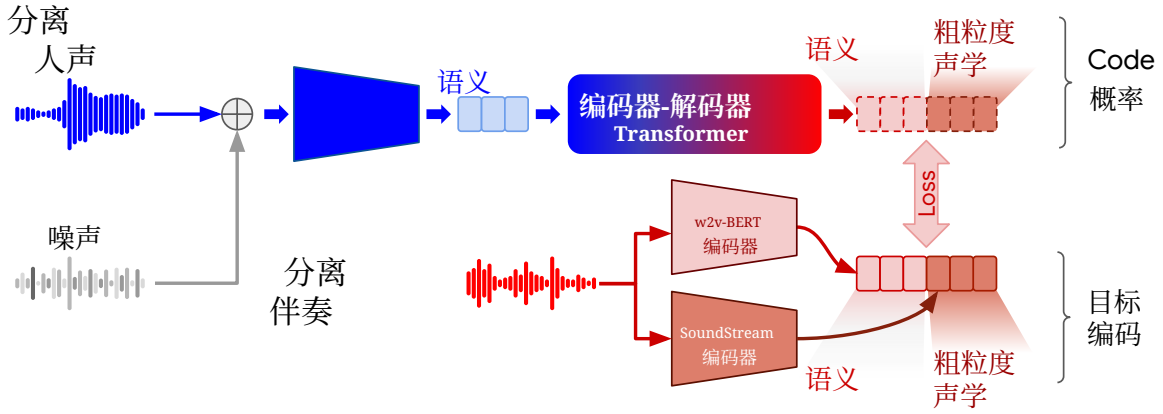


图 2. 我们对 AudioLM（Borsos 等人，2022年）进行改造，使其适用于训练以人声为条件的伴奏“音频到音频”生成模型。训练时，我们分别将源分离人声和伴奏作为输入和目标。我们在输入中加入白噪声，以掩盖源分离人声中残留的伴奏伪影。对于目标，我们沿用 AudioLM 的离散特征化方案，从预训练的 w2v-BERT 模型中提取语义编码（Chung 等人，2021年），并从预训练的 SoundStream 编解码器中提取粗粒度声学编码（Zeghidour 等人，2021年）。我们尝试了多种输入特征化方案——效果最好的模型仅使用加噪人声的语义编码。我们训练 T5（Raffel 等人，2020年），这是一种编码器-解码器 Transformer（Vaswani 等人，2017年），根据输入编码预测目标编码。推理时（图中未展示），我们使用该模型生成粗粒度声学编码，再使用单独训练的模型根据粗粒度编码生成细粒度编码，最后通过 SoundStream 对二者进行解码。

单一的“序列到序列”语言模型，根据人声特征预测伴奏语义编码和粗粒度声学编码（图2）。具体而言，我们通过串接 $\hat{\boldsymbol{x}}$ =特征($\boldsymbol{x}$)构造“输入”序列，并通过串接 $\hat{\boldsymbol{y}} = [\hat{\boldsymbol{y}}_1, \ldots, \hat{\boldsymbol{y}}_{f_k T}] \in (\mathbb{S} \cup \mathbb{V})^{f_k T}$来构造“目标”序列：[语义($\boldsymbol{y}$);粗粒度($\boldsymbol{y}$)], 其中 $f_k = 225$ 表示每秒的编码数量。接下来，我们对其进行建模

$$P_{\theta,\phi}(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{x}}) = \prod_{t=1}^{T f_k} P_\theta(\hat{\boldsymbol{y}}_t \mid \hat{\boldsymbol{y}}_{<t}, E_\phi(\hat{\boldsymbol{x}})), \text{where}$$

$$P_\theta(\hat{\boldsymbol{y}}_t \mid \hat{\boldsymbol{y}}_{<t}, E_\phi(\hat{\boldsymbol{x}})) = \texttt{SoftMax}(D_\theta(\hat{\boldsymbol{y}}_{<t}, E_\phi(\hat{\boldsymbol{x}}))).$$

对于我们的序列编码器（E_ϕ）和解码器（D_θ），我们采用 T5（Raffel 等人，2020年）中的编码器-解码器架构，该架构使用 Transformer（Vaswani 等人，2017 年）。

为了在给定人声输入$\boldsymbol{x}\prime$的情况下将这个训练好的模型用于我们的应用，我们首先对伴奏的语义编码和粗粒度声学编码进行采样 $\hat{\boldsymbol{y}}\prime \sim P_{\theta,\phi}(\hat{\boldsymbol{y}} \mid \hat{\boldsymbol{x}} = \text{Feats}(\boldsymbol{x}\prime))$. 然后，我们丢弃语义编码，并使用阶段 3 根据粗粒度编码生成细声学编码。最后，我们将生成的粗粒度和细声学编码合并为 Combined($\hat{\boldsymbol{y}}\prime$)，并输出 $\boldsymbol{x}\prime + \text{Dec}(\text{Combined}(\hat{\boldsymbol{y}}\prime))$。请注意，我们特意将原始 人声输入 $\boldsymbol{x}\prime$ 与解码后的伴奏 Combined($\hat{\boldsymbol{y}}\prime$) 合并，而不是使用经过编解码器往返处理的人声输入版本。

3.4. 输入人声的特征化

这里我们介绍函数 Feats：$\mathbb{R}^{f_s T} \to \mathbb{X}^{f_x T}$，用于从人声输入中提取离散特征。为了与我们的目标序列保持对称，我们采用一种默认做法，直接沿用乐器目标所使用的特征化方式：$\text{Feats}_{\text{Def.}}(\boldsymbol{x}) = [\text{Sem}(\boldsymbol{x}); \text{Coarse}(\boldsymbol{x})]$，即 $f_x = 225$Hz 和 $\mathbb{X} = \mathbb{S} \cup \mathbb{V}$。

添加噪声。 我们提出在训练和推理过程中都向人声输入添加白噪声。添加噪声的主要目的是掩盖源分离人声中残留的原始伴奏中几乎听不见的伪影（可访问首页脚注中的音频示例页面，聆听这些伪影）。具体而言，我们定义函数 Noise：$\boldsymbol{x} \mapsto \boldsymbol{x} + \boldsymbol{z}$，其中 $\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2)^{T f_s}$，$\sigma$ 是控制噪声振幅的超参数。基于此，我们将 $\text{Feats}(\boldsymbol{x}) = \text{Feats}_{\text{Def.}}(\text{Noise}(\boldsymbol{x}))$ 定义为特征提取函数。

在使用源分离数据进行训练的初步实验中，模型表现出较差的泛化能力。具体而言，输入源分离人声时，模型会输出与原始伴奏极为相似、令人起疑的伴奏；输入分离人声时，则大多输出静音。我们怀疑自己无意中训练模型利用伪影重建原始伴奏。我们假设，通过添加振幅高于伴奏伪影、但低于人声的噪声，可以抑制模型在训练过程中依赖这些伪影，从而有望提升泛化能力。

在第 4.5 节中，我们探讨了多种特征化方式，包括加性噪声以及语义编码与声学编码的不同组合。我们利用 AudioLM 的层级结构和条件化的灵活性，确定了能够

抵御源分离伪影的特征化方式。

4. 实验与结果

这里介绍我们用于训练和评估 SingSong 的实验方案。

4.1. 数据集

训练（源分离）。SingSong的训练集由100万个纯音频源构成，共计46千小时的音乐。我们对该数据集进行预处理，将每个源混音从其原始采样率重采样至44.1kHz（大多数无需重采样），并将立体声混音合并为单声道。接下来，针对预训练SoundStream 和 w2v-BERT 与训练`SingSong`，我们的预处理方式有所不同。对于 SoundStream 和 w2v-BERT，我们将混音重采样至16kHz，提取互不重叠的30秒片段，并使用所有片段进行预训练。训练`SingSong`时，我们进一步进行预处理，从每个混音中提取互不重叠的10秒片段，并将生成的片段输入 MDXNet（Kim 等人，2021年）源分离算法（该算法原生运行于44.1kHz），以提取人声。我们从原始混音中减去源分离人声，得到对应的源分离伴奏。最后，我们将人声和伴奏片段重采样至16kHz，即 SoundStream 和 w2v-BERT 的采样率。

评估（分离人声）。为了评估模型在分离人声上的表现，我们采用 MUSDB18 数据集（Rafii 等人，2017年），其中包含10 小时的专业录音室分轨素材，包含分离的人声和伴奏，来自150 个源混音。我们从每个混音中找出人声存在的、不重叠的10秒片段（定义为人声峰值均方根振幅高于线路电平 -25dB），由此得到训练集中的1232 个片段和测试集中的778 个片段。我们还将原始混音中的每个10秒片段输入 MDXNet，以提取分离的人声，从而能够针对相同的音乐素材，直接比较模型在源分离人声和分离人声上的表现。由于我们不使用 MUSDB18 进行训练，而是使用其训练集进行超参数比较，因此下文将其称为我们的“开发”集。

4.2. 训练数据过滤

为了提高训练数据与目标任务的相关性，我们提出了一种过滤策略，用于过滤掉以下片段：(1) 伴奏处于静音状态；或 (2)人声明显大于伴奏。具体而言，我们会过滤掉伴奏峰值均方根振幅低于 -25dB（相对于线路电平）的片段，或过滤掉人声峰值均方根振幅至少比伴奏高5dB的片段。过滤的目标是使系统倾向于始终输出一些适用于所有输入的伴奏，这比这样一种系统更有用：后者遵循未经滤的训练数据分布，偶尔输出音量较低或静音的伴奏。

4.3. 评估

作为主要的定量评估指标，我们采用由 ´提出的 Fréchet 音频距离（FAD）指标 Kilgour 等人 (2019年)，研究表明，与其他常规自动化指标相比，该指标与人类对音频质量的感知相关性更高。FAD 是音频领域中与图像生成模型基准测试中常用的 FID 指标（Heusel 等人，2017年）相对应的指标。具体而言，FAD 是基于参考音频和生成音频分别经 VGGish 音频分类器（Hershey 等人，2017年）提取的嵌入向量计算得到的两个多元高斯分布之间的 Fréchet 距离（Fr´échet，1957）。

在我们的设定中，我们在 MUSDB18 上计算 FAD（按照第 4.1 节所述进行预处理），并使用真实混音作为参考音频。为了衡量本研究中训练的模型的泛化能力，我们分别将分离人声和源分离人声作为输入，并在两种情况下都将模型输出的伴奏与分离人声混合，生成用于与参考音频计算 FAD 的混音。将源分离人声作为输入的情形并不能代表我们预计在实际应用中会遇到的推理场景——我们特别将以分离人声为输入计算的 FAD 记为 FAD_{i}，将以源分离人声为输入计算的 FAD 记为 FAD_{s}。我们将二者之差（Δ）称为泛化差距。

我们还计算了模型在给定源分离人声和分离人声输入时，对分离伴奏的粗粒度声学编码的负对数似然（NLL）。需要注意的是，在实践中，我们发现 NLL 与我们对质量的主观感知相关性较弱——尽管有些模型在主观评价和 FAD_{i} 上都出现了崩溃，所有模型在训练过程中 NLL 仍大致单调改善（示例见图5 ）。尽管如此，我们仍在附录（表2）中报告 NLL，以便更全面地呈现实验结果。

4.4. 建模超参数

下面概述我们实验采用的默认建模配置。我们采用`t5.1.1.base`配置[3]中实现的默认架构和训练超参数，该配置由`t5x` (Roberts 等人，2022年) 提出，并进行了以下更改：(1) 将丢弃率从0 提高到 0.1；(2) 将相对位置嵌入替换为原始 Transformer 的固定位置编码（Vaswani 等人，2017年），这种编码在处理更长序列时计算效率更高，并且在初步实验中

[3] `t5.1.1.base`: https://bit.ly/3Hy91wp

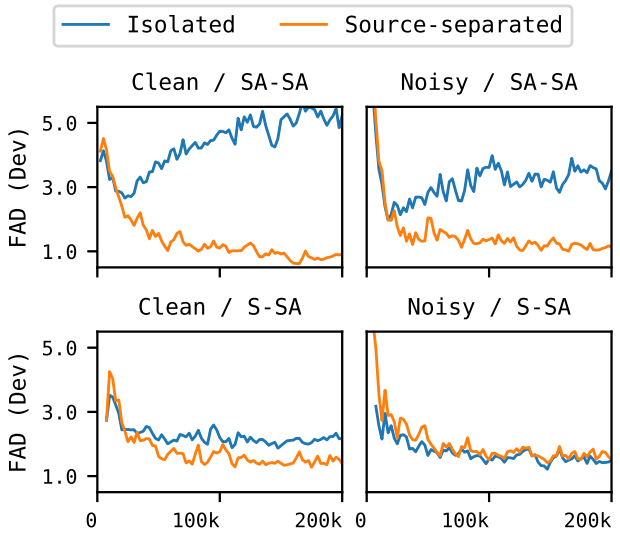


图3。我们尝试对输入人声进行不同的特征化，以提升系统的泛化能力，使其能够从训练期间见到的源分离人声泛化到我们预期用户会提供的分离人声。我们发现，以下两种方法结合起来能够带来良好的泛化效果：(1) 加噪：向人声输入添加白噪声，以掩盖源分离伪影；(2) S-SA：从默认的 AudioLM 特征化（SA-SA）中移除人声的声学编码。具体而言，采用这种特征化进行训练后，在 MUSDB18-dev 上使用分离人声和源分离人声进行评估时，得到的 FAD 几乎相同（右下）。

	方法	FAD_i	FAD_s	Δ
Clean	SA-SA	3.01	1.61	1.39
	S-SA	2.31	**1.14**	1.17
Noisy	SA-SA	2.01	1.64	0.37
	A-A	3.41	3.30	**0.11**
	SA-A	2.81	1.87	0.95
	A-SA	2.01	1.65	0.36
	S-SA	**1.36**	1.17	0.19
	S-SA-XL	**1.28**	**0.96**	0.32

表1. 我们实验的定量评估。FAD 是 MUSDB18-test 上真实混音与模型生成的真实人声和器乐混音之间的 Fréchet 音频距离。模型使用分离人声（FAD_i）或源分离人声（FAD_s）作为输入。由于 FAD_i 更符合预期的使用场景，而 FAD_s 更接近模型的训练方式，因此两者之间存在泛化差距（Δ）。对输入人声添加噪声（加噪）、移除人声声学编码（S-SA）以及增大模型规模（XL）都有助于提升模型对分离人声输入的生成效果（FAD_i）。特征化的完整细节见第 4.5 节。所有实验均使用 MUSDB18-dev 上的 FAD 作为提前停止指标。

我们的任务表现略好。我们在 10秒片段上训练模型 20万步（批次大小为 512），输入和目标的最大序列长度均为 2250。我们根据在 MUSDB18-dev 上计算的 FAD_i，通过提前停止对模型进行正则化。与 AudioLM 相同，我们使用互不重叠的词汇子集表示 4 个粗粒度声学编码维度，词汇量为 5122：其中包含 1024 个语义标记、4096 个声学标记和 2 个哨兵标记（序列起始标记和填充标记——后者未使用）。每个序列索引都唯一对应于该词汇表中的一个特定子集；在采样期间，我们通过从分布中截去无效子集来手动强制执行这一规则。我们采用温度采样进行解码，温度设为 0.85。

4.5. 音频特征化实验

我们针对输入人声和目标伴奏，分别探索了几种不同的音频特征化配置。对于输入人声，我们使用第 3.4 节所述的 Feats 函数，该函数将人声的语义编码和粗粒度声学编码与可选的加性噪声进行拼接。我们实验了两种噪声条件：无噪声（$\sigma = 0$无噪声），以及 加噪，其中 $\sigma = 0.01$，即将人声与经衰减的噪声混合，衰减幅度为 40 dB（通过主观聆听确定，其音量高于大多数伪影但低于人声）。对于目标，我们默认同样将伴奏的语义编码和粗粒度声学编码进行拼接，且不添加噪声，如第 3.3 节所述。我们将这种默认的特征化方式称为 SA-SA，即输入为语义/声学编码，目标也为语义/声学编码。我们还实验了从输入条件中移除粗粒度声学编码（S-SA）。

在图 3 中，我们展示了不同的输入人声特征化会产生不同的泛化特性。无论是加噪 还是 S-SA 配置，似乎都能独立改善泛化能力，即当模型接收源分离人声（如训练时）而非分离人声（在推理时更为现实）时，二者都能缩小模型的泛化差距。两种特征化结合后，基本消除了泛化差距，使训练动态更加稳定，并降低了提前停止的重要性。

我们还尝试了输入和目标的其他特征化方式，以便更好地了解每种特征的影响。我们分别从输入（A-SA）、目标（SA-A）以及输入和目标两者（A-A）中移除语义编码——无法移除目标中的声学编码，因为合成需要这些编码。由于在无噪声条件下训练的两种模型表现不佳，且实验的计算成本较高，我们仅在加噪条件下进行这些额外实验。所有模型的结果见表1。

总体而言，我们的最佳模型（Noisy / S-SA）会向人声输入添加噪声，仅使用人声的语义编码作为条

件信息，并同时使用语义编码和粗粒度声学编码来生成目标伴奏。与 AudioLM 的简单改编模型（Clean / SA-SA）相比，该模型的 FAD_i 提升了 55%。在两项实验（SA-A 和 A-A）中，从伴奏中移除语义编码都会导致性能下降（相较于 SA-SA）。仅对人声使用声学编码（A-SA）与同时使用语义编码和声学编码的效果相近。所有将人声声学编码作为条件信息的模型，在训练步数不足 50k 时，分离人声上的性能都出现了发散。

缩放。我们采用表现最佳的模型配置（Noisy / S-SA），在其他超参数保持不变的情况下，将配置从 t5.1.1.base（2.5 亿个参数）扩展至 t5.1.1.xl（30 亿个参数），并将其称为 S-SA-XL。或许并不令人意外的是，我们发现，与规模较小的 S-SA 相比，扩大模型规模确实提升了定量表现。下文将表现最佳的模型称为 SingSong-Base 和 SingSong-XL。

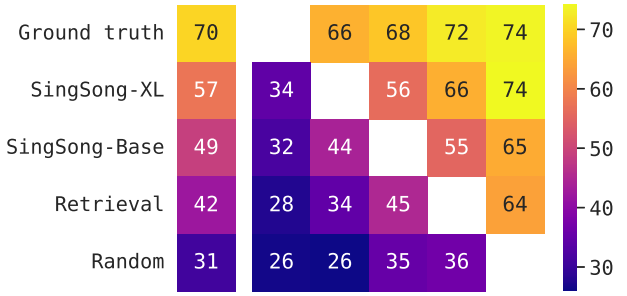


图 *4.* 定性研究的结果：听众对人声固定、伴奏来自两个不同来源（真实值、SingSong 或基线方法）的人声与伴奏混合音频进行成对评判。每一行显示听众选择该系统的伴奏、而非任何其他系统的伴奏（第一列，$N = 800$）以及各个系统的伴奏（其他列，$N = 200$）的百分比。

5. 听测研究

我们开展了一项听测研究，以评估表现最佳的两个模型相对于真实值和基线方法的定性表现。在研究中，听众会听到一对各长 10 秒的人声与伴奏混合音频，其中两段混合音频的人声完全相同，均来自 MUSDB18-test，而伴奏则来自不同来源（真实值、我们的模型或基线方法）。听众需要指出，哪段混合音频中的伴奏与人声在音乐上更协调。我们特别提醒听众不要关注伴奏的音频保真度。

5.1. 基线方法

我们将两种基于检索的基线方法进行比较：一种是从 MUSDB18-dev 开发集中均匀随机地检索一段伴奏片段（随机），另一种则利用真实值混合音频的音乐特征，从 MUSDB18-dev 中检索合适的伴奏，并对其进行调整以进一步改善对齐（检索）。后者的设计与 Songsmith 类似（Simon 等人，2008），但它检索的是伴奏音频，而不是伴奏 MIDI。我们没有直接与 Songsmith 进行比较，因为它没有提供可输入预录人声的 API。

为构建检索基线，我们首先将 MUSDB18-dev 中的每段真实值混合音频输入调性检测系统（Korzeniowski 和 Widmer，2018年），用于估计混合音频处于各个24 种12 个大调、12 个小调）的概率。我们还使用速度检测系统（B¨ock 等人，2015年）估计每段真实伴奏的速度——调性和速度特征均使用这个madmom 库（B¨ock 等人，2016年）计算。随后，对于来自 MUSDB18-test 的输入人声查询，我们使用相同的调性检测系统估计人声的调性概率，并从与该输入人声关联的真实伴奏中检测速度。我们从 MUSDB18-dev 中选取调性概率空间内欧几里得距离最小的伴奏曲目，然后对其进行时间拉伸，使其新速度与输入音频的估计速度相匹配。需要注意的是，由于查询速度是根据关联的伴奏而非输入人声估计的，这条基线获得了现实场景中无法获取的额外信息——我们发现，直接从输入人声估计速度的可靠性太低。

5.2. 结果

所有系统的结果见图4。将我们的最佳系统（SingSong-XL）与最强基线（检索）进行比较时，在 66% 的情况下，听众更喜欢我们最佳系统生成的伴奏。对这些成对判断进行 Wilcoxon 符号秩检验后发现，听众对 SingSong-Base 和 XL生成的伴奏的偏好显著高于最强基线（$p = 0.01$ 和 $p = 2 \times 10^{-6}$，分别）。结果还表明，缩放带来了显著的质量提升——在 SingSong-XL 与56% 的比较中，听众更喜欢前者生成的伴奏，而非 SingSong-Base （$p = 0.01$，采用相同的统计检验）。此外，在与包括真实值在内的任何其他来源进行比较时，听众在57% 的情况下更喜欢我们最佳系统生成的伴奏，而最强基线仅为 42%。请参阅我们的音频示例（见首页脚注）以试听比较结果。

6. 讨论

总体而言，相较于一个能够获得额外信息的强基线，SingSong 生成的伴奏在质量上表现出色。主观来看，SingSong 输出的伴奏往往与输入人声在和声和节奏上有清晰的对应关系。一个有待改进之处是，与打击乐元素相比，生成伴奏中的和声元素通常较弱（无论是音量还是连贯性）。虽然移除人声声学编码对于泛化很重要，但我们认为，仅将人声语义编码用作条件化，可能无法为伴奏生成提供充足的和声背景——我们希望在未来的工作中探索其他输入特征化方法。另一个问题是，AudioLM 声学编解码器的采样率（16kHz）远低于高保真音乐的常规采样率，限制了系统保真度的上限；未来我们希望提高编解码器的采样率。

更长的示例。 在我们的音频示例页面上，我们还分享了最佳模型（SingSong-XL）处理较长（30 秒）人声片段的示例。为获得这些结果，我们以滑动方式将模型应用于 10 秒的人声输入窗口，窗口之间重叠 50%；也就是说，我们先根据前 10 秒的人声生成 10 秒伴奏，然后迭代地生成 5 秒的伴奏片段：每次以 10 秒的人声窗口为条件，并在提示解码器时加入此前生成的 5 秒伴奏，偏移量为$5i$ 秒（首次迭代时为$i = 1$）。从主观角度看，尽管该方法仅以局部上下文（10 秒人声和此前生成的 5 秒伴奏）为条件，但在 30 秒人声输入上仍能实现合理的整体连贯性，并且不同窗口间的风格和配器基本保持一致。

真实场景输入。 在我们的音频示例页面上，我们还展示了系统处理 Vocadito 数据集中的片段的结果（Bittner 等人，2021年）。该数据集包含由新手歌手演唱、使用各种常见麦克风和移动设备录制的人声。由于该数据集缺少真实器乐伴奏，我们没有对其进行评估；不过，与 MUSDB18 相比，它或许更能代表我们预期在现实世界中从用户那里收到的输入。主观而言，Vocadito 上的结果颇有希望，但可能不如 MUSDB18 上的结果理想。在未来的工作中，我们希望探索如何调整用户输入以适配我们的系统，例如让用户像 Simon 等人（2008）那样跟着节拍器演唱，或进行音高修正。

记忆化分析。 我们进行了初步分析，以衡量 SingSong-XL 中的记忆化情况，发现总体而言，无法从我们的模型中逐字提取训练数据。我们采用了类似于Carlini 等人 (2022年所述的方案。具体而言，我们输入训练数据中的人声，并用真实值中的前 k 个语义编码作为提示来提示解码器，然后在使用贪婪采样对剩余的 $250 - k$ 个乐器语义编码进行解码时，报告与真实值完全匹配的数量。对于每组 $k \in \{0, 25, 50, 125\}$，我们在 *8*千次试验中均未发现完全匹配。对于 $k = 225$，我们仅发现 4 次完全匹配（0.05%）。

未来工作。 除了我们已经提到的方向，未来值得探索的途径还包括进一步研究源分离与音频到音频伴奏生成之间的关系。首先，音乐源分离已经能够提取人声和器乐之外的其他声部，例如鼓、贝斯和钢琴。未来，我们希望训练伴奏模型，使其能够根据任意其他声部生成任意声部。源分离与伴奏生成之间也可能存在互利共生的关系。源分离质量的提升（伪影更少）可能缩小伴奏系统的泛化差距，而源分离灵活性的提高（支持更多乐器类别）则可能带来新的伴奏能力（例如风笛伴奏）。同样，伴奏方面的改进也有助于为源分离生成合成但逼真的训练数据。例如，可以将 SingSong 应用于大量未经混音的歌唱录音，生成对齐的（混音、人声）配对数据。

伦理考量。 尽管音乐生成技术为扩大人们参与音乐创意活动的机会提供了契机，但也带来了深远的风险——音乐生成技术可能从根本上重塑音乐文化，并重新定义利益相关者之间的经济关系（Detweiler 等人，2022年）。更具体地说，在所有乐器中，歌声或许最能唤起人们对个人身份的联想。我们构建的音乐生成系统（1）要求用户主动参与（歌唱）来生成音乐，（2）在输出中保留个人身份特征。我们希望，该系统能够扩大人们参与音乐创意活动的机会，同时避免从头生成音乐或模仿现有歌手身份的系统所带来的弊端。目前系统的一个风险是，用户无法控制伴奏的流派和风格——我们的系统可能已经对用户文化身份与某些文化身份所关联的音乐流派之间的相关性存在偏见。未来，我们希望像 Agostinelli 等人（Agostinelli 等人（2023）那样，根据流派、风格或文本描述来控制系统，从而通过让用户明确控制这些高层属性来降低这一风险。

参考文献

Agostinelli，A.、Denk，T. I.、Borsos，Z.、Engel，J.、Verzetti，M.、Caillon，A.、Huang，Q.、Jansen，A.、Roberts，A.、Tagliasacchi，M.、Sharifi，M.、Zeghidour，N. 和 Frank，C.《MusicLM：根据文本生成音乐》。*arXiv:2301.11325*，2023年。

Bittner, R. M., Pasalo, K., Bosch, J. J., Meseguer-Brocal, G. 和 Rubinstein, D. vocadito：一个包含独唱人声、 f_0、音符和歌词标注的数据集。*arXiv:2110.05580*，2021年。

B¨ock, S., Korzeniowski, F., Schl¨uter, J., Krebs, F. 和 Wid- mer, G. madmom：一个新的 Python 音频与音乐信号处理库。载于*ACM* 多媒体国际会议，2016年。

Borsos,Z.,Marinier, R., Vincent, D., Kharitonov, E., Pietquin, O., Sharifi, M., Teboul, O., Grangier, D., Tagliasacchi, M. 和 Zeghidour, N. AudioLM：一种用于音频生成的语言建模方法。*arXiv:2209.03143*，2022年。

B¨ock, S., Krebs, F. 和 Widmer, G. 基于循环神经网络和共振梳状滤波器的精准速度估计。载于*ISMIR*，2015年。

Caillon, A. 和 Esling, P. RAVE：一种用于快速、高质量神经音频合成的变分自编码器。*arXiv:2111.05011*，2021年。

Carlini, N., Ippolito, D., Jagielski, M., Lee, K., Tramer, F. 和 Zhang, C. 量化神经语言模型中的记忆化。*arXiv:2202.07646*，2022年。

Chung, Y.-A., Zhang, Y., Han, W., Chiu, C.-C., Qin, J., Pang, R. 和 Wu, Y. w2v-BERT：结合对比学习与掩码语言建模的自监督语音预训练方法。载于*IEEE* 自动语音识别与理解研讨会（*ASRU*），2021年。

Detweiler, C., Coleman, B., Diaz, F., Dom, L., Donahue, C., Engel, J., Huang, C.-Z. A., James, L., Manilow, E., Mc- Croskery, A., Pedersen, K., 等. 重新定义音乐中的关系。*arXiv:2212.08038*，2022年。

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A., 和 Sutskever, I. Jukebox：一种音乐生成模型。*arXiv:2005.00341*，2020年。

Dieleman, S., van den Oord, A., 和 Simonyan, K. 逼真音乐生成的挑战：大规模建模原始音频。发表于*NeurIPS*，2018年。

Donahue、C.、McAuley、J. 和 Puckette、M. 对抗音频合成。载于*ICLR*，2019年。

Engel, J., A$_{\text{grawa}}$ l, K. K., Chen, S., Gulrajani, I., Donah ue, C., 和 Roberts, A. GANSynth：对抗神经音频合成。发表于*ICLR*，2019年。Engel, J., Hantrakul, L., Gu, C., 和 Roberts, A. DDSP：可微数字信号处理。发表于*ICLR*，2020年。Fr´echet, M. 关于两个概率分布之间距离的研究。《科学院周刊》，1957年。Goel, K., Gu, A., Donahue, C., 和 R´e, C. It' s raw! 使用状态空间模型进行音频生成。发表于*ICML*，2022年。Grachten, M., Lattner, S., 和 Deruty, E. BassNet：一种用于条件生成贝斯吉他音轨、带有习得交互式控制的变分门控自编码器。《应用科学》，2020年。

Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.-Z. A., Dieleman, S., Elsen, E., Engel, J. 和 Eck, D. 使用 MAESTRO 数据集实现因子化钢琴音乐建模与生成。发表于*ICLR*，2018年。

Hawthorne, C., Jaegle, A., Cangea, C., Borgeaud, S., Nash, C., Malinowski, M., Dieleman, S., Vinyals, O., Botvinick, M., Simon, I. 等。使用 Perceiver AR 进行通用长上下文自回归建模。发表于*ICML*，2022年a。

Hawthorne, C., Simon, I., Roberts, A., Zeghidour, N., Gard- ner, J., Manilow, E. 和 Engel, J. 使用频谱图扩散进行多乐器音乐合成。发表于*ISMIR*，2022年b。

Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., 等。用于大规模音频分类的卷积神经网络架构。载于*ICASSP*，2017年。

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. 和 Hochreiter, S. 使用双时间尺度更新规则训练的生成对抗网络会收敛到局部纳什均衡。*NeurIPS*，2017年。

Kilgour, K., Zuluaga, M., Roblek, D. 和 Sharifi, M. Fr´echet 音频距离：一种用于评估音乐增强算法的无参考指标。载于*INTER-SPEECH*，2019年。Kim, M., Choi, W., Chung, J., Lee, D. 和 Jung, S. KUIELab-MDX-Net：一种用于音乐解混的双流神经网络。载于*MDX*研讨会论文集，2021年。

Korzeniowski、F. 和 Widmer、G.《使用卷积神经网络进行与流派无关的调性分类》。载于 *ISMIR*，2018年。

Lattner, S. 和 Grachten, M. 使用学习到的节奏互动模式对鼓轨生成进行高层控制。载于*WASPAA*，2019年。

Mehri, S.、Kumar, K.、Gulrajani, I.、Kumar, R.、Jain, S.、Sotelo, J.、Courville, A. 和 Bengio, Y. SampleRNN：一种无条件的端到端神经音频生成模型。ICLR，2017年。

Paiement, J.-F., Eck, D. 和 Bengio, S. 概率旋律和声。发表于加拿大人工智能会议，2006年。

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P. J. 等。探索统一文本到文本转换器的迁移学习极限。机器学习研究期刊，2020年。

Rafii, Z., Liutkus, A., St¨oter, F.-R., Mimilakis, S. I. 和 Bittner, R. MUSDB18——一个用于音乐分离的语料库。2017年。

Roberts, A., Chung, H. W., Levskaya, A., Mishra, G., Bradbury, J., Andor, D., Narang, S., Lester, B., Gaffney, C., Mohiuddin, A., Hawthorne, C. 等。使用 T5X 和 SeqIO 扩展模型与数据。*arXiv:2203.17189*，2022年。

Simon, I.、Morris, D. 和 Basu, S. MySong：面向声乐旋律的自动伴奏生成。载于 计算机系统中人为因素特别兴趣小组，2008年。

van den Oord, A.、Dieleman, S.、Zen, H.、Simonyan, K.、Vinyals, O.、Graves, A.、Kalchbrenner, N.、Senior, A. 和 Kavukcuoglu, K. WaveNet：用于原始音频的生成模型。*arXiv:1609.03499*，2016年。

van den Oord, A.、Vinyals, O. 和 Kavukcuoglu, K. 神经离散表征学习。载于*NeurIPS*，2017年。

Vaswani, A.、Shazeer, N.、Parmar, N.、Uszkoreit, J.、Jones, L.、Gomez, A. N.、Kaiser, Ł. 和 Polosukhin, I. Attention Is All You Need。载于*NeurIPS*，2017年。

Wu, Y.-K., Chiu, C.-Y., 和 Yang, Y.-H. JukeDrummer：通过 Transformer 向量量化变分自编码器生成有条件、节拍感知的音频域鼓伴奏。载于 *ISMIR*，2022年。Yeh, Y.-C., Hsiao, W.-Y., Fukayama, S., Kitahara, T., Genchel, B., Liu, H.-M., Dong, H.-W., Chen, Y., Leong, T., 和 Yang, Y.-H. 使用三和弦进行自动旋律和声编配：一项比较研究。*arXiv:2001.02360*，2020年。Zeghidour, N., Luebs, A., Omran, A., Skoglund, J., 和 Tagliasacchi, M. 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，2021年。

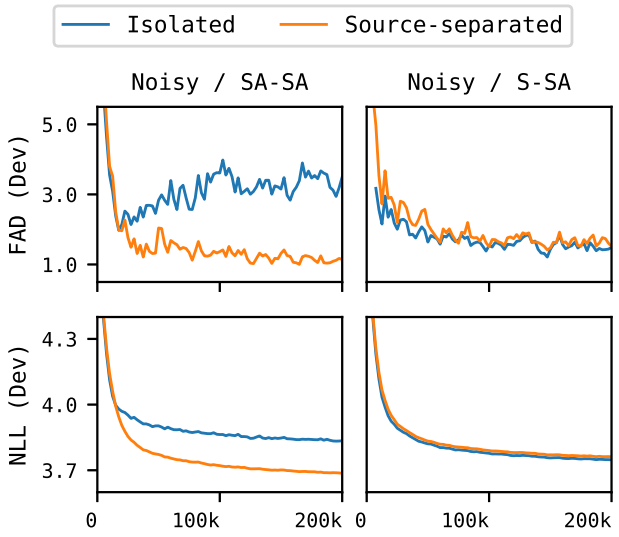


图5。与弗雷歇音频距离（FAD）相比，我们发现负对数似然（NLL）并不能很好地反映这项任务的主观表现。

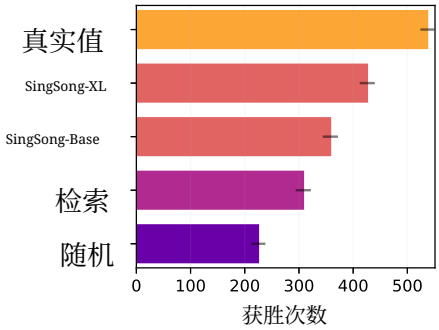


图 6。人类听众研究的结果。获胜次数表示在并排比较中某个模型胜过另一模型的次数，误差线表示均值的标准差。

A. 补充实验结果

在图 5中，我们比较了两项实验（SA-SA 和 S-SA 在加噪 条件下）的 FAD 和负对数似然（NLL），输入分别为分离人声和源分离后的人声。我们注意到，在两项实验中，分离人声的 NLL 都单调下降，而其中一项实验中，分离人声的 FAD 却出现发散。此外，听取不同检查点的音频示例后，我们发现，相比 NLL，我们的主观判断更多时候与 FAD 一致。因此，本研究将定量评估和分析重点放在 FAD 上。表2报告了分离人声的 NLL（NLL_s）和源分离后的人声输入的 NLL（NLL_s）。

我们还报告了另外两项实验的结果，两者均采用加噪/SA-SA 条件：（1）NoFilt 取消了第 4.2 节所述的数据集过滤；（2）RelPos 采用相对位置嵌入——这是 T5 的默认设置（Raffel 等人，2020年）——而非我们在其他实验中使用的原始 Transformer 的绝对位置编码（Vaswani 等人，2017年）。我们发现，与使用过滤和绝对位置编码相比，取消过滤和使用相对位置嵌入都会导致 FAD 变差$_i$。绝对位置编码的计算效率也更高，尤其是在序列较长时。

B. 补充听力研究详情

在图6中，我们以模型在所有两两比较中相对于其他系统的原始“胜出”次数，展示了听众研究结果的另一种视图（$N = 800$），误差线表示估计胜出比例的标准差。

图 7 展示了参与者在听众研究中进行评分评估时看到的界面。我们明确要求评分者不要关注音频质量，而应着重判断人声与伴奏音轨的契合度。他们评估的音频片段是输入人声（上方）与我们研究的五种方法之一的混合音频。具体而言，评分者会听到一段分离的人声演唱音频，并看到我们测试的两种伴奏系统各自生成的一段混合音频。随后，评分者需要回答：

> 你觉得哪段伴奏听起来像是专为这段歌唱创作的？不要关注音频质量，而要着重判断哪段音乐与歌唱最契合（节奏和和声相似）。

B.1. 检索基线的其他细节

这里我们报告基线检索系统的其他实现细节。首先，为了尽量减少时间拉伸产生的伪影，我们试图将人声与伴奏的估计速度之比保持在区间 $[0.5, 2.0]$ 内。因此，当该比值小于 0.5 时，我们将其乘以 2.0；当该比值大于 2.0 时，则将其除以 2.0。其次，人声查询始终恰好为 10 秒，因此我们从 MUSDB18-dev 检索集中移除了 4 条短于 20 秒的音轨；20 秒是可能的最大时间拉伸比。

	方法	步骤	FAD_i	FAD_s	Δ	NLL_i	NLL_s
Clean	SA-SA	23k	3.01	1.61	1.39	3.84	3.79
Clean	S-SA	148k	2.31	**1.14**	1.17	3.72	3.71
Noisy	SA-SA	18k	2.01	1.64	0.37	3.92	3.89
Noisy	NoFilt	20k	3.33	2.30	1.03	3.93	3.88
Noisy	RelPos	43k	2.24	1.07	1.17	3.80	3.79
Noisy	A-A	18k	3.41	3.30	**0.11**	4.03	3.99
Noisy	SA-A	43k	2.81	1.87	0.95	3.94	3.89
Noisy	A-SA	48k	2.01	1.65	0.36	3.85	3.79
Noisy	S-SA	148k	**1.36**	1.17	0.19	**3.71**	3.70
	S-SA-XL	150k	**1.28**	**0.96**	0.32	**3.47**	**3.47**

表 2。 我们实验的扩展定量评估。FAD 是 Fr´echet Audio Distances，用于衡量 MUSDB18-test 上真实值混音与由真实值人声和器乐组成的混音之间的距离；后者由分离人声（FAD_i）或源分离人声（FAD_s）作为模型输入生成。由于 FAD_i 更符合预期的使用场景，而 FAD_s 更接近模型的训练方式，因此两者之间存在泛化差距（Δ）。NLL 是以源分离人声（NLL_s）和分离人声（NLL_i）作为模型输入时，给定输入计算伴奏声学编码的负对数似然。对输入人声添加噪声（加噪）、移除人声声学词元（S-SA）、增大模型规模（XL）均有助于提升分离人声的生成效果（FAD_i）。特征化的完整细节见第 4.5 节。所有实验均使用 MUSDB18-dev 上的 FAD 作为提前停止指标，“Steps”表示提前停止前的训练步数。

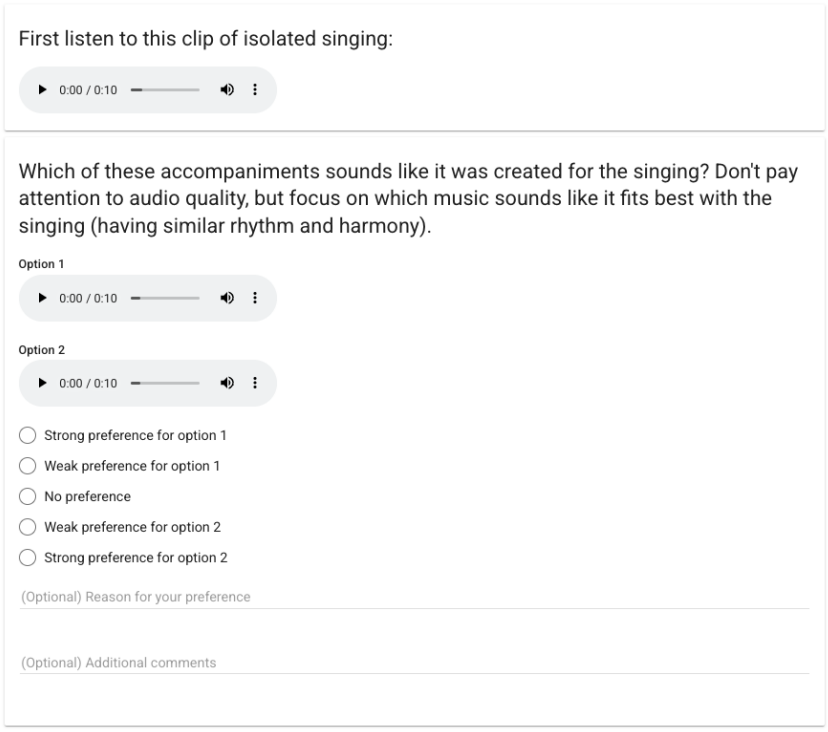


图 7. 听众研究参与者用于进行评估的评分界面截图。标记为选项 1 和选项 2 的音频片段（位于提示下方）是人声输入与我们重点关注的五种方法之一生成的伴奏混合而成：[真实值、SingSong-XL、SingSong-Base、检索、随机]。