音乐分离研究中可复现性的代价：频带分割循环神经网络的复现

Paul Magron、Romain Serizel、Constance Douwes

arXiv:2603.09187v1 [cs.SD] 10 Mar 2026

摘要——**音乐源分离是从音乐歌曲中分离出乐器音轨的任务。尽管近期进展显著，但架构与训练方案日趋复杂的趋势加剧了可复现性问题。频带分割循环神经网络（BSRNN）模型在这方面颇具前景，因为它在公开数据集上取得了接近最先进水平的结果，且训练所需资源合理。遗憾的是，由于其完整代码不可用，复现并非易事。本文通过大量实验，尽可能贴近原论文复现 BSRNN，从而对这一可复现性问题进行批判性反思。我们的贡献有三方面。首先，本研究对模型设计与训练流程提出若干见解，为未来可能的改进提供启示。特别是，由于我们未能复现原始结果，我们探索了更多变体，最终得到优化后的 BSRNN 模型，其性能大幅优于原始模型。其次，我们从方法论与实践两个角度讨论可复现性问题。我们特别强调，若完整流程可用，本可节省大量时间与能源成本。第三，我们公开发布代码与预训练模型，以促进可复现研究。我们希望本研究有助于提升音乐分离社区对可复现研究重要性的认识，并推动更加透明、可持续的实践。**

索引词—**音乐声源分离、频带分割循环神经网络、复现研究、可复现研究、可复现性危机。**

I. 引言

音乐源分离（MSS） [1] 旨在提取组合起来构成一首歌曲的各个乐器音轨。这项技术可用于多种场景，例如，为人工耳蜗用户自动混音 [2], [3], 、生成卡拉OK／伴奏音轨 [4], [5], ，或执行下游音乐信息检索任务 [6], [7]。

得益于深度学习的出现，音乐源分离在过去十年中取得了显著的性能提升。大量模型、架构和优化策略被提出，并呈现出向更深、更复杂网络发展的趋势。早期方法侧重于频谱图域建模 [8], [9],，而较新的模型则依赖于复值频域 [10], [11] 或时间/混合域网络 [12]。各种神经网络架构已被采用，包括循环 [10], [13],卷积 [12]‚[14], 或 Transformer [11], [15]网络。社区驱动的举措，如音乐解混挑战赛（MDX） [16]‚[17] ，推动了这一对性能的追求，从而如今在受控基准测试上衡量时，分离质量已十分出色。

尽管取得了这些进展，但从可复现研究的角度来看，近期的大多数方法仍存在若干缺陷。例如，最新 MDX 挑战赛的获胜方案 [17] 都是模型组合，由于依赖多个基础架构，训练起来可能十分繁琐。其他例子包括通过在私有数据集上进行微调而获得的最佳性能模型，这使得复现几乎不可能 [15]。最后，一些系统需要如此强大的计算能力，以至于在只能获得其中一小部分资源时，重新训练所需的时间会长到难以接受 [11]。这些因素加剧了应用机器学习许多领域所面临的可复现性危机 [18]‚[19],，其中也包括音乐信息检索 [20], [21]。

在当前最先进的音乐源分离方法中，频带分割循环神经网络（BSRNN） [10] 从可复现性的角度来看颇具前景。事实上，它的性能优于在公开可用的 MUSDB18-HQ 数据集 [22], 上训练和评估的更新方法，因此是一个强有力的基线，能够与竞争方法进行公平比较。它还是一种单一架构，相较于模型组合更易于训练和微调；与更新近的方法相比，其报告的训练时间和所需硬件也相当合理 [11]。遗憾的是，据我们所知，目前没有可用的完整流程实现来复现论文中报告的结果，其中包括数据准备代码以及详细的训练和评估脚本。此外，尽管非官方实现包含有用的资源，但其性能显著低于原论文（见第二节-B）。

本研究探讨音乐源分离研究领域中的可复现性问题 [21], [23]。尽管与药理学 [24] 或生态学 [25], 等其他领域相比，应用机器学习中的复现研究较为少见，但要评估整个社区层面的进展，回顾并反思以往工作仍然十分重要。为此，我们力求尽可能忠实地复现 BSRNN，以贴近原论文 [10]。具体而言，我们详细介绍训练和评估方案，以阐明复现过程中遇到的困难。我们开展了大量实验，研究各种参数、设计选择和训练策略。由于未能达到原论文报告的性能，我们实现了若干可能弥合这一差距的变体。最终，我们得到一个优化后的 BSRNN模型，其性能优于论文结果。通过这项深入调查，我们得以对可复现性问题及其各种代价进行批判性反思。总而言之，

P. Magron 和 R. Serizel 任职于法国南锡的洛林大学、法国国家科学研究中心、Inria、LORIA（邮编：F-54000；电子邮箱：paul.magron@inria.fr、ro-main.serizel@loria.fr）。Constance Douwes 任职于法国马赛的 Centrale Med、艾克斯-马赛大学、法国国家科学研究中心、LIS（电子邮箱：constance.douwes@lis-lab.fr）。

我们的工作主要贡献如下：

1) 我们进行了大量实验，以评估 BSRNN 基础架构的参数及其变体。这些实验为模型设计和训练流程提供了若干见解，并揭示了音乐源分离未来可能的改进方向。此外，尽管本研究的主要目标并非进一步推动最先进水平，但我们的实验最终得出了经过优化的 BSRNN 变体，其表现大幅优于原论文结果。

2) 通过深入讨论，我们表明，重视可复现性有助于加快音乐源分离研究，并提高能效。我们特别关注本项目的能源足迹，因为鼓励机器学习各个领域采取更可持续的实践至关重要 [26]，[27]，，音乐信息检索社区也包括在内 [28]。

3) 为实现这一目标并促进可复现研究，我们公开发布了代码[1] 和预训练模型。[2] 这些资源轻量且开放，可作为社区的替代方案，与表现最佳的音乐源分离系统相比颇具实用价值 [11]。

本文其余部分安排如下。第二节概述近期的音乐源分离方法，指出其可复现性问题，并阐述开展这项复现研究的动机。第三节介绍 BSRNN 模型，以及我们考察的一些架构变体。第四节介绍实验方案，第五节详述实验结果。第VI节随后讨论可复现性的成本。最后，第七节总结全文。

II. 概述与动机

本节旨在说明开展此类复现研究的动机。为此，我们概述了最先进音乐源分离（MSS）系统的可复现性问题，其性能报告于表 I 中。为使本节简洁，我们将在第四节介绍数据集和评估指标，并在附录A详细说明这些数据的收集过程。

A. 模型比较

我们先来看表 I 的上半部分，其中报告的结果仅使用公开可用的 MUSDB18-HQ 数据集。尽管我们列出 CWS-PResUNet 的 [29] 性能以供参考，但其性能显著低于较新的模型，因此不将其纳入复现研究。KUIELab-MDX-Net [30] 和报告中的 Hybrid Demucs [12] 均为模型集成／模型包，这意味着它们结合了多个子模型的输出。这些子模型可以采用不同的基础架构，也可以采用同一架构的不同实例，并通过不同的超参数进行调优和训练。因此，它们的优化过程尤其繁琐，Hybrid Demucs 的作者对此也有说明：

> “虽然适合用于竞赛，但如此复杂的模型确实会妨碍轻松复现其所达到的性能。” [12, 第10]页

[1]github.com/magronp/bsrnn [2]zenodo.org/records/13903584

表 I MUSDB18-HQ 测试集上最先进的音乐源分离模型的分离性能（分块信号失真比，单位：分贝）。

	人声	Bass	鼓	其他	平均值
在 MUSDB18-HQ 上训练					
CWS-PResUNet [29]	8.92	5.93	6.38	5.84	6.77
KUIELab-MDX-Net [30]	8.97	7.83	7.20	5.90	7.47
Hybrid Demucs [12]	8.35	8.43	8.12	5.65	7.64
HT Demucs [15]	7.93	8.48	7.94	5.72	7.52
BSRNN [10]	10.01	7.22	9.01	6.70	8.24
SIMO-BSRNN [31]	9.73	7.80	10.06	6.56	8.54
BS-RoFormer [11]	10.66	11.31	9.49	7.73	9.80
DTTNet [32]	10.12	7.45	7.74	6.92	8.06
TFC-TDF UNet v3 [33]	9.59	8.45	8.44	6.86	8.34
使用额外数据进行训练					
Hybrid Demucs [12]	8.75	9.13	9.31	6.18	8.34
HT Demucs [15]	9.37	10.47	10.83	6.41	9.27
BSRNN [10]	10.47	8.16	10.15	7.08	8.97
BS-RoFormer [11]	12.72	13.32	12.91	9.01	11.99

此外，尽管 HT Demucs [15] 比 Hybrid Demucs 更先进，但在未使用额外数据进行训练时，其性能反而较差。这表明，虽然 HT Demucs 同样适用于竞赛或私人使用，但对于研究论文而言，它可能并不是一个合适的比较参考。

DTTNet [32]，、SIMO-BSRNN [31]，和 BS-RoFormer [11]都是基于 BSRNN模型构建的变体。一方面，DTTNet 是一个轻量级版本，其性能略低于 BSRNN。另一方面，BS-RoFormer 将 Transformer 融入 BSRNN，尽管其性能最佳，但模型规模过大，超出了我们的计算资源所能支持的训练范围。[3] 最后，SIMO-BSRNN 改进了 BSRNN 的架构和训练流程，但仍存在 BSRNN 的可复现性缺陷。因此，在这些变体中，BSRNN 似乎是最适合复现的模型。

最后，TFC-TDF UNet v3 [33] 的性能与 BSRNN 相近，但后者对音乐源分离社区的影响更大。事实上，BSRNN 推动了后续发展（例如上述变体），也被用于影视音频分离 [34] 或语音增强 [35]等其他任务。因此，在这些最先进的音乐源分离模型中，BSRNN 似乎是最佳的复现对象。

请注意，表格下半部分展示的是使用额外（私有）数据进行训练和/或微调所得到的结果。报告这些结果仅仅是为了说明微调特定模型能够带来多大提升（例如，BS-RoFormer 的提升约为 2.2 分贝），但这些结果本质上无法复现。此外，由于不同论文使用的额外私有数据集各不相同，这些结果既不能用于评估某个模型的最佳／最大潜力，也无法用于实际比较不同模型。

[3]原论文 [11]中的每个 BS-RoFormer 模型都使用 16 块 Nvidia A100（80 GB）GPU 训练了 4 周。若将这一设置应用于我们最大的集群（见第四节-D2），全局批次大小至少需要缩小 4 倍。因此，针对数据集中的每种乐器（共四种），单次训练运行都需要约 4 个月。

B. 问题设定

现在让我们讨论 BSRNN 特有的可复现性问题。作者分享了 MDX 挑战赛中用于定义模型的一些代码。[4] 然而，目前没有可用的官方实现能够复现论文中报告结果所需的<样式 id='2'>完整流程</样式>。这包括数据预处理代码、包含优化器和调度器参数的详细训练脚本，以及用于处理完整歌曲的推理流程等评估函数。然而，这些内容对于复现任何基于深度学习的系统的性能都至关重要。此外，即使是前述已发布的代码，也无法直接使用，因为它无法按照原论文 [10]计算损失（见第四节-D1）。据我们所知，SIMO-BSRNN 也存在同样的问题：没有提供官方模型定义，而且还需要更多实现细节。

得益于其在研究社区中的受欢迎程度，已有一些非官方的 BSRNN 实现，但它们并未完全解决上述问题。例如，A. Amatov 的实现[5] 性能远低于原始实现（人声轨道为 6.67 分贝，而原始实现为 10.01 分贝）。C. Landschoot 将该实现适配到了另一个音乐源分离数据集，[6] 还有另一个项目[7] 利用 BSRNN 进行语音增强， [35]，因此，相应的处理流程并不适用于原始任务。

因此，目前尚无公开实现能够复现 BSRNN 原论文中的结果。这促使我们开展 BSRNN 复现研究，并发布代码作为完全独立的实现，同时提供训练好的模型，以推动进一步的研究和开发。

III. 模型与变体

本节将简要介绍原始 BSRNN 模型。随后，为缩小我们的实现与论文结果之间的性能差距，我们提出了几种架构变体。原始模型及其变体如图 1 所示。

A. 原始 *BSRNN* 架构

BSRNN 以混合信号的复数短时傅里叶变换（STFT）$\mathbf{X} \in \mathbb{C}^{F\times T}$作为输入，其中$F$ 和 T 分别表示频带数和时间帧数，并预测目标声源的 STFTS $\in \mathbb{C}^{F\times T}$。实际上，复数 STFT 通常表示为实部和虚部的堆叠，因为这种实值张量可以很容易地由神经网络处理。BSRNN 由以下三个模块组成。

首先，频带分割模块将输入的短时傅里叶变换分解为 K 个带宽可变的子带频谱图。随后，这些频谱图被投影到维度为 N的深层潜在空间中，然后将得到的深子带特征重新堆叠成维度为 $N\times K\times T$的全频带张量。该频带分割方案针对每个声源单独设计并优化，因此产生了大小各异的乐器专用模型。

接着，序列与频带建模模块依次应用两个残差网络，其基本结构由组归一化、双向长短期记忆网络（LSTM）和全连接层组成。序列与频带循环神经网络分别沿着 T 时间帧和 K 频带进行处理。这种双路径方法 [36] 通过捕捉时间和频率上的依赖关系，充分利用音频/音乐信号的结构。将此类循环神经网络堆叠 R次，以构成更深的网络。

最后，掩码估计 模块会将前一个模块的输出拆分为多个子带，并将每个子带的特征输入多层感知机（MLP），以预测子带掩码。MLP 使用隐藏层大小 $\mu \times N$，其中 $\mu = 4$。随后，将子带掩码组合成全频带掩码**百万能量** $\in \mathbb{C}^{F\times T}$，并将其与输入短时傅里叶变换相乘，得到源估计：$\mathbf{S} = \mathbf{M} \odot \mathbf{X}$，最后通过逆短时傅里叶变换（iSTFT）将其转换回时域。

B. 变体

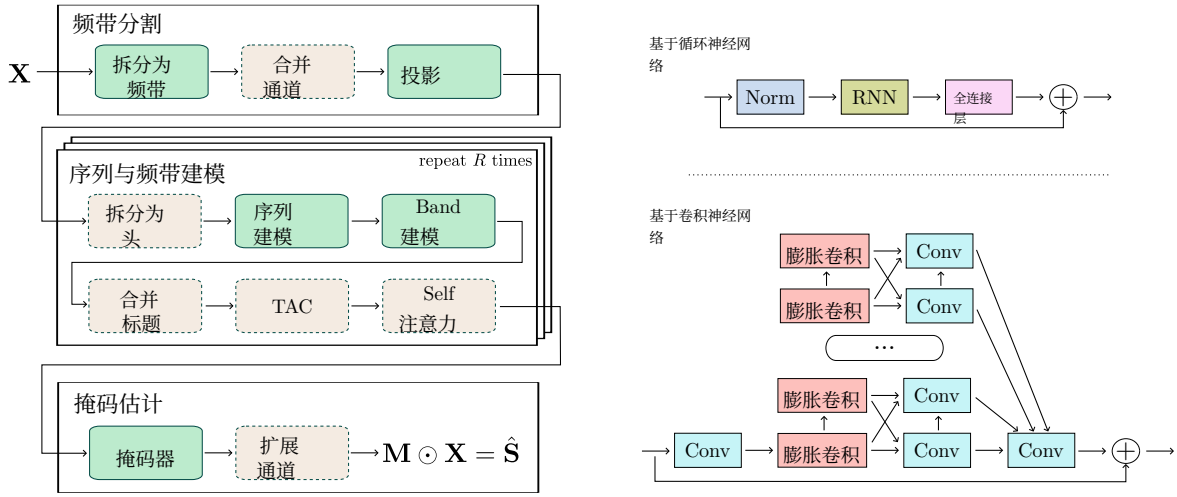
这里介绍我们提出的架构变体，并为清晰起见，重点阐述关键概念。完整的实现细节请参阅我们的代码。

1) 立体声建模： 根据设计，BSRNN 是一种单通道模型，会独立处理（立体声）输入混合信号的两个通道，就好像它们是来自不同歌曲的两条单通道流。忽略跨通道信息可能会限制系统性能，而独立处理左、右通道则会造成不必要的计算负担。

为缓解这一问题，我们提出了一种 BSRNN 立体声变体，它基于对频带分割和掩码估计模块进行简单修改。我们将立体声输入混合信号视为一个 2 通道短时傅里叶变换 $\mathbf{X} \in \mathbb{C}^{2\times F\times T}$，并首先将其分割为子带。然后，将每个子带频谱图的左、右通道拼接起来，并共同投影为维度为 N 的深子带特征。序列与频带建模模块保持不变，而掩码器的多层感知机的输出数量加倍，分别对应估计源的两个通道。下文将这一策略称为朴素立体声模型。它对应于图 1-(A)中的“合并通道”和“扩展通道”模块。

或者，我们提出采用变换-平均-拼接（TAC）模块。该模块最初专为语音分离设计 [37], ，近期也用于 SIMO-BSRNN变体中的音乐源分离 [31]。简而言之，TAC模块首先通过全连接层对多通道输入表示进行变换；然后对各通道进行平均，并通过第二个全连接层处理；最后，将第二层的输出与第一层变换后的表示拼接，再输入第三个全连接层。由于该TAC模块能在通道间共享信息，因此可实现立体声感知建模。我们将其置于序列与频带建模 [31], 之后，以及全连接层的

[4]gitlab.aicrowd.com/Tomasyu/sdx-2023-music-demixing-track-starter-kit [5] github.com/amanteur/BandSplitRNN-PyTorch [6] github.com/crlandsc/Music-Demixing-with-Band-Split-RNN [7] github.com/sungwon23/BSRNN



(A) BSRNN 架构　　(B) 残差网络

图 1. BSRNN 及其变体 (A)。虚线轮廓的模块表示原架构中不存在的变体。残差网络 (B) 序列/频带模块所基于要么是循环神经网络（如原始模型中那样），要么是膨胀卷积。

激活函数可以是双曲正切函数（TanH），如SIMO-BSRNN [31]，所用；也可以是参数化整流线性单元（PReLU），如原始TAC论文 [37]所用。

2) 替代层：我们研究序列与频带建模模块中长短期记忆网络的替代层。初步实验表明，门控循环单元与长短期记忆网络的结果相近（如其他研究 [34]所述），而将每个循环神经网络替换为简单的单层卷积神经网络（CNN）会导致性能大幅下降，尽管由于内存限制降低，这些网络的训练速度要快得多。然而，这种方法相当朴素，因为此类卷积神经网络无法妥善建模具有长期依赖的序列数据。因此，我们建议使用一叠膨胀卷积神经网络，因为它们在源分离方面已展现出良好前景，并且由于其更大的感受野而可与循环神经网络相媲美 [12], [38]。我们实现了Wang [38], 的架构，得到一个全卷积模型，记为频带分割卷积神经网络（BSCNN），如图1-(B)所示。

3) 自注意力：自注意力机制已在语音增强和分离等许多涉及序列数据的任务中展现出良好前景 [39], [40]。特别是，它被用于 TFGridNet [40], ，该模型与 BSRNN 具有相似之处，因为它通过双路径式架构学习跨频带与时间帧的依赖关系。在 TFGridNet 中，自注意力机制被观察到能带来显著的性能提升，而对模型规模和计算负担的影响可忽略不计。我们在每个序列与频带建模模块中纳入 N_a 个注意力头，编码维度为 E_a ，并采用 ESPnet 工具箱 [41]中的实现。

4）多头机制：Chen 等人 [32] 提出用所谓的“改进”模块替代序列与频带建模模块。这些模块归结为将维度为 $N \times K \times T$ 的输入特征拆分为 H 个注意力头，每个注意力头的维度为 $H \times N/H \times K \times T$。随后，循环神经网络在各个注意力头上并行运行，使用N/H 的较小隐藏维度，而不是 N，从而大幅减少（冗余）参数的数量。这对应于图 1 中的“拆分为注意力头”和“合并注意力头”模块 -(A)。

IV. 实验方案

本节描述我们的实验方案。由于它与原始 BSRNN 论文 [10], 大体相似，我们主要关注考虑了变体的方面，或需要特别注意和澄清的细节。

A. 数据集与数据处理

我们在所有实验中使用公开可用的 MUSDB18-HQ 数据集 [22]。它包含 150 首采样率为 44100 Hz 的立体声歌曲，每首配有混合音频及其对应的四个分离音源：人声、贝斯、鼓和其他。这些歌曲分别被划分为 86、14 和 50 个音轨，用于训练、验证和测试。短时傅里叶变换使用长度为 2048 个样本的汉宁窗计算，跳跃步长为 512。

按照原论文的方法，我们每个训练周期生成 20,000 个训练样本。首先，我们使用音源活动检测器（SAD）对训练曲目进行预处理，去除静音区域。然后，按以下方式即时生成训练样本：

(i) 我们从不同歌曲中随机选择声源。[8](ii) 我们为每个声源随机截取时长为 3 s 的数据块。(iii) 我们对每个数据块应用随机缩放因子，使其能耗处于 $[-10, 10]$ 分贝范围内。(iv) 我们以 0.1 的概率丢弃每个数据块，以模拟静音声源。(v) 我们将数据块相加，形成混合音频。

我们还考虑一种替代的训练数据生成流程，不使用 SAD 预处理，也不随机丢弃数据块，但采用与 Open-Unmix (UMX) 相同的数据增强方式 [8], ，即以

[8]初步实验并未发现仅在训练开始时打乱曲目与在每个训练周期重新打乱曲目之间存在差异。我们采用前一种方式，因为它稍快一些。此外，请注意，尽管这一过程会产生不协调的混合音频——这种现象被称为刺耳噪声 ——但研究表明，它能有效提升音乐源分离的性能 [42]。

0.5 的概率随机交换通道，并使用 0.25 到 1.25 之间的线性增益重新调整数据块的能耗。

B. 模型配置

为了加快原型开发，大多数实验使用小型模型，即 $N = 64$ 和 $R = 8$；而原论文使用 $N = 128$ 和 $R = 12$，在本文中对应于标记为大型或优化型的模型。除非另有明确说明，所有其他超参数均按照原论文 [10], 设定。

C. 指标

我们使用信号失真比（SDR）评估声源分离质量，单位为分贝（越高越好） [43]。我们考察 SDR 的以下两种变体：

- 片段SDR（uSDR）等于在完整歌曲上计算的基本信噪比。我们报告歌曲间的平均值，与 MDX 挑战赛 [16], [17],中将其用作评估指标的做法一致。
- 数据块 SDR（cSDR）通过计算时长为 1 秒的数据块的 SDR 中位数得出。我们按照信号分离评测活动（SiSEC）的指南 [44], 报告歌曲间的中位数，并使用 museval 工具箱 [45]计算 cSDR。

D. 训练

1) 损失：BSRNN 使用组合损失$\mathcal{L} = \mathcal{L}_{\text{time}} + \mathcal{L}_{\text{STFT}}$ 进行训练，该损失由一个时域项组成：

$$\mathcal{L}_{\text{time}} = |\text{iSTFT}(\mathbf{S}) - \text{iSTFT}(\hat{\mathbf{S}})|_1, \tag{1}$$

以及一个短时傅里叶变换域项：

$$\mathcal{L}_{\text{STFT}} = |\mathbf{S}_r - \hat{\mathbf{S}}_r|_1 + |\mathbf{S}_i - \hat{\mathbf{S}}_i|_1, \tag{2}$$

其中，$|.|_1$ 表示 ℓ_1 归一化层，且下标 r 和 i分别表示实部和虚部。

2) 优化器和硬件：原论文使用 Adam 算法训练模型，初始学习率为 $\lambda = 10^{-3}$，批量大小为 2，并行使用 8 个 GPU，从而得到全局批次大小 $B = 16$。遗憾的是，我们无法使用足够多（大型）的 GPU 来达到相同的全局批次大小。因此，我们调整学习率，以保持相同的有效学习率 λ/B，从而确保梯度下降步长 [46]相近。学习率每两个训练轮数衰减为原来的 0.98 倍，并采用最大梯度范数为 5 的梯度裁剪。

小型模型使用 4 块 Nvidia RTX 2080 Ti（11 GB） GPU 进行训练，但带自注意力的鼓组模型因内存限制而使用 4 块 Nvidia Tesla T4 （ 15 GB ） GPU 进行训练。大模型使用 2 块 Nvidia Tesla L40S（45 GB）GPU 进行训练。

3) 监控：原论文中训练最多进行 100 个训练轮数 [10],，我们观察到这在大多数情况下并不足以实现收敛，因此我们将该数值设为 200。此外，作者采用早停，“当连续 10 个训练轮数内未找到最佳验证时” [10, IV-A],，但并未明确说明所监控的是哪一项指标（损失或信号失真比）。默认情况下，我们选择监控 uSDR，因为计算 cSDR 较为耗时，并将耐心值保持为 10 个训练轮数。

E. 用于评估的完整歌曲推理

在评估阶段（即验证或测试阶段），由于内存限制，模型无法一次性处理整首歌曲。因此，需要将歌曲划分为较小的（重叠）片段，输入分离模型，再通过重叠相加（OLA）策略拼接估计得到的数据块，以还原完整源信号。

在 BSRNN 论文中，评估片段时长为 3 秒，跳跃步长为 0.5 秒，但并未说明具体的 OLA/重建技术（即加窗、裁剪/拼接等）。于是，我们实现了一种简单的流程，假设采用矩形窗，并将缩放因子设为确保完美重建所需的值。另一种做法是将歌曲划分为时长 10 秒、重叠率为 10% 的片段，再使用线性淡化器拼接估计得到的数据块，以平滑片段边缘 [12]。

初步实验表明，在验证阶段，推理流程不会影响模型选择，因为无论验证集 uSDR 存在何种轻微的绝对差异，最终选出的最佳模型都相同。因此，我们在验证时采用线性淡化器，因为它速度快得多。不过，推理流程会影响测试得分：对于 BSRNN [10], ，人声轨道的得分差异最高可达 0.3 分贝；对于 TFC-TDF UNet v3 [33]，也观察到了类似情况。尽管比较模型时常常忽略这一点，我们将在第 V-C 节中讨论。

F. 能耗监控

我们监测了本研究中所考察的各种模型训练所消耗的能耗。为此，我们通过 codecarbon 工具箱 [47]跟踪训练期间的能耗（以千瓦时计）。由于这种方法往往会低估能耗，我们还采用 Green Algorithms 在线计算器 [48], 的方法，该方法根据硬件规格估算能耗（我们按每 8 GB 内存 3 W 的功耗计算）。研究表明，与 codecarbon [49] 相比，这种方法得出的结果更接近实际功率计读数。根据计算平台的建议，我们采用 1.5 的电源使用效率系数。[9] 请注意，如第四节-D 所述，我们的各种模型使用不同的 GPU 进行训练，这会影响能耗，因此可能影响比较结果 [50]。尽管如此，我们仍采用这一设置，因为它可以让我们充分利用现有硬件。

[9]https://intranet.grid5000.fr/stats/indicators

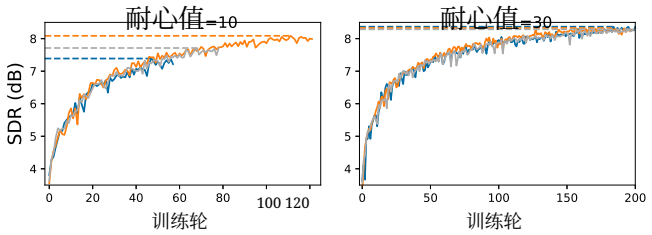


图2. 基础模型在人声轨道上的验证集 uSDR 随训练轮数的变化情况，耐心值为 10（左）或 30（右）。每种颜色对应一次不同的运行，虚线表示各次运行的最佳 uSDR。

我们还估算了整个项目的总能耗，包括所有其他模型变体（本文未报告）、初步及额外实验、原型开发和推理。为简便起见，我们报告使用 Green Algorithms 计算器计算出的项目总能耗。

V. 结果

本节介绍我们的实验结果。由于篇幅将过于冗长，本文并未报告所有已进行的实验。更多详情请参阅配套文档，[10] 我们也鼓励感兴趣的读者使用我们的代码进行实验，以开展进一步研究。

A. 验证结果

我们在表 II 中报告了所有变体的最佳模型验证集 uSDR 和训练能耗。每一行都说明了与基础模型（第 1 行所示）的差异，该模型作为各变体的比较基准。

1) 基础模型：首先考虑一个基础模型，它对应于 BSRNN 的小型版本（参见第 IV-B 节）。作为初步实验，我们使用三个不同的随机种子训练该模型，并在图 2（左）中展示人声轨道随训练轮数变化的验证集 uSDR（其他声源也得到类似结果）。我们观察到，所有运行都呈现相同趋势，但最佳 uSDR 存在一定差异。这种差异是由 uSDR 的不稳定性造成的，而这种不稳定性源于不同的初始随机种子 [51]，最终导致训练在不同的轮数停止。

为缓解这一问题，我们将耐心参数提高到 30，并在图 2（右）中展示结果。我们观察到平均最佳 uSDR 有所提高，但更重要的是，其方差大幅降低。因此，提高耐心值能有效地继续训练，并降低随机种子对最佳 uSDR 的影响。然而，其代价是训练轮数以及随之而来的能耗增加到原来的 2.3 倍。

基于这些发现，人们既可以将随机种子像任何其他超参数 [52],一样进行调优以降低方差，也可以像上文那样对多次运行取平均值/提高耐心值。尽管如此，为了使我们的实验在时间和能源成本上保持高效，并与原论文保持一致，除非另有说明，我们报告单次运行的结果，耐心值为10 明确说明。例外是基础模型，我们报告三次初始运行的平均值，因为这些运行我们已经完成。

2) 训练参数：
梯度累积 我们也可以不按第四节-D2 所述调整学习率，而是在执行下降之前跨步累积梯度，从而人为增大全局批次大小。第 2 行的结果表明，两种策略表现相近。在下文中，我们调整学习率，因为我们观察到在训练更大的模型时这种方式更为稳定。

监控准则 默认情况下，早停通过监控 uSDR 来执行（见第四节-D3），我们也可以改为监控验证损失。两种方法得到的结果相近（参见第 3 行），但我们选择前者，因为它允许训练持续更多训练轮数，有利于确保收敛。

损失我们通过分别最小化损失的各项来进行训练（见第四节-D1）。有趣的是，所有损失域（参见第 1、4 和 5 行）都取得了相似的结果，这与先前研究强调时域训练重要性的结论相悖 [53]。一种解释是，STFT 域损失 (2) 分别处理实部和 虚部。这可能会隐式地强制实现相位一致性 [54], [55]，使其等同于时域损失。

3) 原始架构参数：
STFT参数 我们采用窗口长度为 4096 个采样点、跳跃步长为 1024 个采样点的 STFT，因为这种设置在音乐源分离模型中很常见 [8], [12]。如第 6 行所示，这会导致较差的结果，我们认为这是由于时间分辨率降低，且增加的频率分辨率无法弥补这一点。事实上，后者影响不大，因为频带划分方案和潜在空间中的投影（维度固定）都在网络的早期阶段进行。这也解释了为什么鼓的性能下降比低音轨更明显：打击乐事件在时间上具有局部性，因此需要更精细的时间分辨率才能得到恰当建模。同样，使用更大的 6144 点窗口 [32] 会得到更差的结果。

掩码器大小 我们提出将 $\mu = 2$ 设置为1，以减小多层感知机掩码器的大小（相较于默认的 $\mu = 4$，见第三节-A）。虽然这会降低鼓和其他声部的性能，但会显著改善贝斯的结果，从而获得相近的平均性能（参见第7行）。进一步将 μ 降至1会导致性能更加明显地下降。

大模型 使用原始模型规模（$N = 128$ 和$R = 12$）时，与基础模型相比，平均uSDR提高了1.1分贝（第8行）。然而，该模型尚未完全收敛（鼓音轨除外），因此我们在提高耐心值后继续训练，使性能进一步提升了0.2分贝（第9行）。不过，考虑到延长训练导致能耗增加了54%，这项提升似乎非常有限。尽管如此，该模型在测试集上的性能仍低于原始结果（见第V-C节），这促使我们开展接下来的实验。

4) 架构变体：
立体声模型 我们采用朴素方法进行立体声建模，结果导致性能下降，但低音轨除外，

[10]github.com/magronp/bsrnn/blob/main/docs/analysis.md

表 II 各模型性能对比：最佳模型的验证集 uSDR（单位：分贝）、参数总数（单位：百万，“-”表示与上一行数值相同），以及训练所有声源的预计能耗（单位：千瓦时）。

		验证集 uSDR（分贝）					# 参数（百万）	能耗（千瓦时）	
		人声	Bass	鼓	其他	平均值		codecarbon	绿色算法
1	基础模型： $N=64$, $R=8$	7.7	6.1	9.7	4.8	7.1	32.3	127	168
	训练参数								
2	梯度累积	8.0	5.8	9.6	4.9	7.1	-	129	170
3	使用损失进行监控	7.5	6.4	9.3	4.8	7.1	-	120	159
4	损失域：时间	7.9	6.1	9.4	4.9	7.1	-	116	153
5	损失域：短时傅里叶变换	7.9	6.4	9.6	4.9	7.2	-	131	173
	原始架构参数								
6	短时傅里叶变换：窗口=4096，步长=1024	7.3	5.9	8.7	4.4	6.6	37.1	58	92
7	掩码因子 $\mu=2$	7.9	6.8	9.4	4.4	7.1	20.6	110	151
8	大模型： $N=128$, $R=12$	9.2	7.3	10.3	5.8	8.2	146.7	230	321
9	带耐心值的大模型=30	9.5	7.8	10.3	6.3	8.4	-	354	495
	立体声模型								
10	朴素方法	7.7	6.6	8.4	4.0	6.7	37.1	78	122
11	朴素方法，带有 $\mu=8$	7.9	6.1	8.7	4.3	6.7	81.1	87	140
12	采用 TanH 激活函数的 TAC	7.6	6.0	9.6	4.3	6.8	34.7	117	154
13	采用 PReLU 激活函数的 TAC	7.9	6.5	10.0	4.7	7.3	34.7	128	167
	架构变体								
14	BSCNN	7.3	5.9	9.0	4.2	6.6	29.7	113	153
15	注意力： $N_a=1$, $E_a=8$	7.7	7.4	10.4	4.8	7.6	33.0	151	199
16	注意力： $N_a=2$, $E_a=16$	8.2	7.7	10.4	4.9	7.8	33.2	157	224
17	多头模块， $H=2$	7.6	5.5	9.1	4.0	6.6	22.0	91	137
	数据生成								
18	静音目标（而非所有声源）	7.9	6.6	9.5	4.4	7.1	32.3	110	146
19	不使用语音活动检测；采用 UMX 类数据增强	8.2	6.9	9.5	5.3	7.5	-	135	179
	优化后的模型								
20	注意力、耐心值 = 30、不使用语音活动检测	10.1	9.1	10.9	6.7	9.2	149.9	426	593
21	+ TAC	10.2	10.2	11.3	6.9	9.6	164.1	508	711

第 10 行的结果也证实了这一点。弥补这一差距的一种方法是增大掩码器大小（$\mu=8$），以补偿输出数量的增加，因为必须恢复两个通道。虽然这在一定程度上减轻了鼓音轨和其他音轨的性能下降，但低音轨的结果反而更差（参见第 11 行）。此外，若进一步增大 N 和 R 以匹配“大型”配置，模型在训练时会变得大到难以承受。与 SIMO-BSRNN变体 [31], 类似，采用 TAC 模块也会导致性能下降（见第 12 行）。然而，如 TAC 论文 [37], 最初所提议的那样，以 PReLU 替换 TanH 激活函数后，模型性能优于基础模型，且低音轨和鼓音轨的提升更为显著（第 13 行）。

BSCNN 我们在第 14 行报告了所提出的 BSCNN 模型的性能。此前，我们进行了初步实验，以调整卷积超参数（膨胀层数量、卷积核尺寸等）。我们观察到，尽管 BSCNN 训练速度更快、模型略轻，但其性能不及基于循环神经网络的网络。对卷积参数进行更精细且针对不同声源的调优（例如，为频带建模和序列建模采用不同参数）可能会提升性能，这留待未来研究。

自注意力 如第 15 行和第 16 行所示，加入自注意力头是有益的，但其他声部除外。具体而言，这种小型鼓组模型加入注意力后，性能与不含注意力的大模型相近（参见第 8 行）。因此，与完全以 Transformer 取代循环神经网络、规模更大且计算开销高得多的模型相比，这种方法是一个有趣的替代方案 [11]。

多头模块 第 17 行的结果显示，多头模块对大多数声源的表现较差，人声轨道除外。设置 $R=4$ [32] 或进一步增加 H 会导致性能明显下降，不过模型会变得轻量得多。

5）数据生成：生成数据 [10,IV-A-2], 时，我们不再随机丢弃每个数据块来模拟静音声源，而是只丢弃目标声源，这提高了低音轨的性能（参见第 18 行）。此外，正如第 19 行所示，不执行语音活动检测（SAD）并应用 UMX 类数据增强，进一步提升了平均性能。这表明我们的语音活动检测实现仍有改进空间，因为据称这种预处理能显著改善分离效果。不过，其影响究竟有多大尚不明确，因为我们没有通过专门的实验进行评估 [10]。

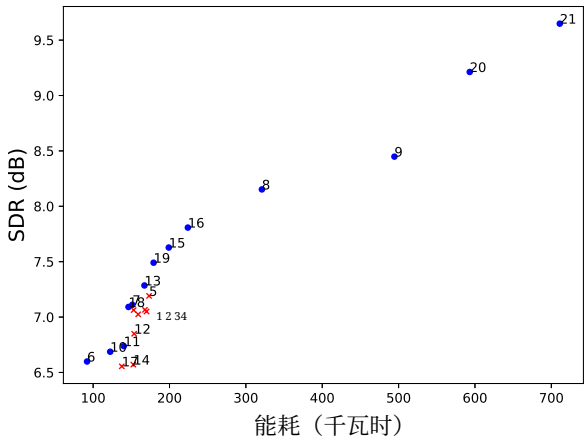


图3. 模型变体的性能与训练能耗（使用 Green Algorithms 计算器估算）之间的关系，其中每个数字标记对应表 II 中的行号。蓝色圆点标出了帕累托意义上的最优模型。

6）优化后的模型：我们使用这一替代数据生成流程，训练了一个大模型（$N = 128$ 和 $R = 12$），并采用自注意力头（$N_a = 2$ 和 $E_a = 16$），将耐心值提高到 30。根据第 20 行所示的结果，该模型优于我们之前的大模型（第 9 行），代价是参数数量适度增加。进一步加入用于立体声建模的 TAC 模块（采用 PReLU 激活函数）后，得到一个优化后的模型，性能额外提升了 0.4 分贝（第 21 行）。

B. 能耗

我们在表 II 最后几列报告了各模型训练所消耗的能耗。请注意，第 1 行对应三次运行的平均值（见 V-A1节）。我们观察到，两种能耗估算技术在模型间能耗相对变化方面的表现总体相似。Green Algorithms 计算器得出的数值高于 codecarbon，这与以往研究一致（见例如 IV-F节）。

大多数小型模型变体的能源成本相近，但架构更轻、内存需求更低或通常收敛更快的模型（第 6、10 和 17 行）除外；这些特点都能降低能耗。相反，带有注意力机制的模型（第 15 和 16 行）能源成本较高，是因为它们需要更多训练轮数才能收敛。这凸显了在进行模型选择时，同时考虑性能和能耗的重要性。例如，第 18 行的模型虽然性能与基础模型相近，但训练能耗更低，因此是一个值得考虑的替代方案。我们还在图3中展示了性能与能耗之间的关系，并标出了帕累托意义上的最优点 [56], ，也就是说，不存在其他点能同时取得更高的信号失真比和更低的能耗。这样便可排除那些存在性能更好且成本更低替代方案的变体，例如使用 TAC 和 TanH 激活函数的模型（第 12 行）与基础模型（第 1 行）相比。

总体而言，训练表中讨论的所有模型共耗费4,808千瓦时，相当于单个最佳模型的能耗成本的约7倍。乍看之下，这一比例似乎尚属合理，但我们也估算了该项目的总能耗，详情见IV-F节。总计为23兆瓦时，超过训练最佳模型能耗的32倍，也相当于基础模型能耗的136倍。为了便于理解，这相当于欧洲约15人一年的用电量。[11] 如此巨大的代价表明，有必要系统地报告能耗并采用更可持续的开发实践，其中包括提高研究的可复现性，因为如果当时有完整且文档齐全的代码，部分能源成本本可避免（见第VI节）。

C. 测试结果

MUSDB18-HQ 测试集上的分离结果见表 III。首先，我们将基于线性推子的推理流程与基于重叠相加的流程进行比较，后者与原始 BSRNN 论文中的方法类似（参见第IV-E节）。使用该重叠相加流程时，改变跳跃步长会使人声轨道的结果相差 0.1 分贝，这与原始结果 [10, 表 II]一致。平均而言，重叠相加和渐变器技术的结果相近，但使用 0.5 秒跳跃步长的重叠相加会使推理时间增加到原来的 6 倍。这证实了我们的线性淡化器有助于加快推理。使用优化模型也能得出类似结论。

我们发现，我们的实现平均比原始测试结果 [10]低 0.5 分贝。这促使我们进一步改进处理流程，以更好地匹配原始结果。表 III 还列出了优化模型 oBSRNN 的性能，该模型在第V-A6节中有介绍。该模型显著提升了我们的 BSRNN 实现性能，弥合了性能差距；在 uSDR 和 cSDR 方面，它甚至分别比原论文结果高出 0.6 和 1.2 分贝。因此，这一优化模型是 BSRNN 的一个有趣替代方案，因为它可公开获取，且结果更好。

关于 SIMO-BSRNN 的说明 我们研究了在 SIMO-BSRNN 中实现的其他建模变体 [31]。这些变体包括更精细的频带拆分方案、在各声源间共享频带拆分模块和序列与频带建模模块，以及一种将相邻频率用作上下文的掩码器。为简洁起见，我们不在此展示相应的实验，详情请参阅补充文档。我们优化后的版本在表 III 中记为 oBSRNN-SIMO，其性能显著优于 SIMO-BSRNN，实际上与 BS-RoFormer 不相上下（参见表 I 的结果）。由于该模型达到了最先进的音乐源分离质量，如果目标是实现最佳性能，可以考虑使用它。

VI. 讨论

这项复现研究在诸多方面都提供了深刻见解。值得注意的是，它扩展了原始 BSRNN 论文的分析，考察了多种参数，例如架构相关参数（如掩码器的多层感知机大小）、训练相关参数（如各项损失的贡献

[11]https://ec.europa.eu/eurostat/statistics-explained/index.php?title=电力_和_热力_统计。

表 III MUSDB18-HQ 测试集上的声源分离性能（uSDR 和 cSDR，单位：分贝）。

	人声		Bass		鼓		其他		平均值	
	uSDR	cSDR	uSDR	cSDR	uSDR	cSDR	uSDR	cSDR	uSDR	cSDR
论文结果 [10]	10.04	10.01	6.80	7.22	8.92	9.01	6.01	6.70	7.94	8.24
我们的实现										
渐变器	9.16	9.14	6.51	7.72	8.55	8.07	5.44	5.68	7.42	7.65
重叠相加，步长 = 1.5 秒	9.10	-	6.47	-	8.33	-	5.44	-	7.34	-
重叠相加，步长 = 0.5 秒	9.18	-	6.60	-	8.43	-	5.53	-	7.44	-
优化后的模型										
oBSRNN	9.78	9.81	8.38	9.85	10.28	10.31	5.85	6.31	8.57	9.07
oBSRNN-SIMO	10.73	10.66	8.32	9.73	10.74	10.98	6.80	7.78	9.15	9.79

），以及数据相关参数（如 SAD 预处理）。明确报告此类结果，即使有时是负面的，也能让其他研究人员受益，因为这可以避免他们耗费大量成本进行（重复）研究。此外，这些详细结果也为模型缩减方面的潜在改进铺平了道路。例如，缩小掩码器大小或采用我们朴素的立体声方法，都能得到性能相近但更轻量的低音模型。或者，使用带注意力机制的小型鼓组模型，其性能可与更大的变体媲美，同时显著轻量化。我们的实验还表明，不能将其他论文中的架构模块原封不动地拿来复用。例如，膨胀卷积 [38]，或多头模块 [32]，在相应论文中被证明有效，但它们是与其他架构模块配合使用，并集成在不同的处理流程中。同样，SIMO-BSRNN 模型 [31] 中采用的变体也不能直接照搬，因为其中只有掩码器上下文是单独评估的。

尽管我们的分析主要关注以信号失真比数值表示的性能结果，但根本性的可复现性问题并不止于此，这也呼应了音乐信息检索领域此前关于该议题的研究 [20], [21]。首先，结果之间的差异目前尚无解释，而其成因可能是本研究所考察的某一种或多种因素，包括但不限于这些因素。由于整个处理流程没有可用的官方实现，我们无法保证自己没有出错，也无法保证不同实现之间不存在重大差异。其次，复现过程本身极其耗时，阻碍了研究进展。事实上，如果代码（连同训练流程、超参数，最终还包括权重）能够公开发布，那么整个项目的实现（包括原型开发和调试）以及为调整超参数而进行的大量训练都可以省去。这样一来，后续研究便能在该项目的基础上更快推进，而不必重新实现。

该项目在深度学习方面的特性带来了额外的可复现性成本。事实上，深度学习研究对环境的影响相当显著，应朝着系统性监控并减少环境足迹的方向努力 [26],[27], ，而不可复现的研究会加剧这一问题，正如我们在第五节 B 部分的结果所概述的那样。公平地说，部分能耗成本源于我们自身的实现错误，例如导致训练运行中断或重复，但我们认为，大多数音乐／音频研究人员并非机器学习或编程专家，因此这些陷阱很可能普遍存在。此外，我们的研究显示，其能耗与作业数量比与类似项目开发阶段的典型情况相符。例如，先前一项研究的作者 [27] 报告称，其整个项目的电力成本约为训练单个模型的 2000 倍；他们还报告总共运行了近 5000 个作业，其中许多作业都失败了。总体而言，此类项目的能耗成本相当可观，在进行模型选择时应系统地报告并予以考虑。此外，如果当时能提供代码、文档并进行适当的超参数选择，本研究的环境足迹本可大幅降低，因为许多反复试错的实验都可以避免。

最后，深度模型的性能很大程度上取决于硬件，因为硬件会影响训练方案。一方面，鉴于（全局）批量大小会影响训练速度和收敛，进而影响最终性能，要适应现有硬件，就必须考虑并仔细调整适应策略（例如缩放学习率或进行梯度累积），这会带来相当大的额外实验负担。另一方面，对于计算能力有限的公共机构而言，可复现性在实践中可能根本无法实现。例如，如第二节所述，若要适当调整训练设置以适配我们最大的集群，训练单个 BS-RoFormer 源模型 [11] 大约需要 4 个月，这使得任何深入的实验研究都无法开展。硬件能力上的这种差异限制了可用于复现研究的模型选择，而在研究中对新系统与竞争模型进行基准测试时，也应将这一点纳入考量。

七、结论

在本文中，我们探讨了音乐分离研究中的可复现性问题。我们概述了最先进的音乐源分离模型存在的若干缺陷，并提出对 BSRNN模型进行复现和深入分析。我们实现并发布了一个功能完整的处理流程，其中包括性能可与最先进系统相媲美的优化后的模型。我们还讨论了与该项目相关的各种成本，尤其是能耗方面的成本。本研究与其他机器学习领域的相关工作相呼应，例如推荐系统 [18] 或语言处理 [19], ，并且可能适用于语音增强和分离等其他任务，因为这些任务与音乐源分离一样容易面临类似风险。

除了关注性能之外，本工作的核心贡献在于提醒人们，可复现性是科学事业的一个基本方面。我们鼓励同行采用开放研究实践，尤其是发布附带完善文档的代码 [21], ，但我们也承认，由于版权问题或公司政策，有时可能无法做到这一点。在这种情况下，我们谨慎建议，应主要根据相关论文的方法论或理论价值对其进行评判；数值性能结果则应更加审慎地报告，尤其是在进行模型比较时。此类实践将促进更加透明、可靠且具有成本效益的研究。

VIII. 致谢

所有计算均在 Grid5000（https://www.grid5000.fr）测试平台上完成。该平台由一个法国科学兴趣小组提供支持，由 Inria 托管，成员包括法国国家科学研究中心、RENATER、数所大学及其他组织。感谢 BSRNN 论文的作者 Jianwei Yu 解答了我们有关实现的一些问题。我们还感谢 Christopher Landschoot 就他个人的非官方 BSRNN 实现进行了富有成效的讨论，并感谢 Stefan Uhlich 就 UMX 和 MDXchallenge 展开了富有见地的讨论，这些讨论有助于改进我们的工作。

附录A
音乐源分离模型性能报告

本附录介绍了从表 I 收集测试结果的方法。这有助于避免在对比论文时可能出现的歧义或不一致。

首先回顾一下，第四节-A 中介绍的 MUSDB18 数据集包含一个高质量版本 MUSDB18-HQ [22], 和一个压缩版本，本文称为 “nonHQ” [57]。这两个版本并存，虽然为社区提供了更大的灵活性，但也可能使结果报告变得复杂。

- 在其原论文中，ResUNet 模型 [58] 是在 MUSDB18 非高质量版上评估的。然而，它常与其他在 MUSDB18-HQ 上测试的模型一起出现在分离基准中 [12, 表 3], ，这多少有些令人困惑；有时还被报告为在 HQ测试集上计算得出 [32, 表 3], ，这是错误的。我们未将该模型列入表 I，因为其在 MUSDB18-HQ 上的性能不可得。

- UMX [8] 官方仓库[12] 报告了两个数据集上的结果。一篇后续论文报告了在 MUSDB18 非高质量版上的性能 [59, 表 1], ，但这些数值对应于官方仓库中的 MUSDB18-HQ 性能。由于这一混淆，以及其整体性能相较于更新方法偏低，我们未纳入该模型。

- 类似地，随 KUIELab-MDX-Net 模型 [30] 一同发表的原论文报告的是 MUSDB18-nonHQ 上的性能，而这些相同的数据随后又被 HT Demucs 论文中的{v1}报告为 MUSDB18-HQ 上的性能 [10]。接着，BSRNN 论文 [10]报告了这两个数据集上的性能，因此我们从该来源提取高质量版结果，并将其列于表 I 中。
- 最后，我们未纳入其他流行模型，如 Spleeter [9], 或 D3Net [60],，因为据我们所知，它们的性能仅在 MUSDB18 非高质量版上有所报告。

另一个造成混淆的因素是存在多个模型变体。例如，Hybrid Demucs 的原始配套论文 [12] 报告了基础模型和优化后的模型集成的性能。后续论文则只报告其中一种 [15]或另一种 [10], [11], ，既未明确说明具体是哪一种，也未提供理由。表 I 中我们报告的是优化后的模型集成所对应的结果，因为它的信号失真比（SDR）最高。HT Demucs 也同样如此，我们报告原论文 [15], 中的最佳结果，但训练时使用额外数据的模型的平均信号失真比除外（我们将 9.20 替换为实际平均值 9.27 分贝）。

最后，其余模型（CWS-PResUNet、TFC-TDF UNet v3，以及 BSRNN 和各个变体）的结果均摘自相关原始论文。

这段说明提醒我们，引用其他论文的结果时务必格外谨慎。我们尤其强调，应说明所使用的数据集版本，以及所选用的模型变体。这样才能确保比较恰当且公平，尤其是在进行模型排名时。

[12]github.com/sigsep/open-unmix-pytorch

参考文献

[1] E. Cano, D. FitzGerald, A. Liutkus, M. D. Plumbley, 和 F. Stöter, 《音乐源分离：入门介绍》, *IEEE* 信号处理杂志，第36卷，第1期，第31–40页，2019年1月。[2] J. Pons, J. Janer, T. Rode, 和 W. Nogueira, 《利用源分离算法重新混音，以改善人工耳蜗使用者的音乐体验》, 美国声学学会杂志，第140卷，第6期，第43384349页，2016年12月。[3] S. Tahmasebi, T. Gaje.cki, 和 W. Nogueira, 《面向人工耳蜗使用者音乐重新混音的实时音频源分离算法的设计与评估》, 神经科学前沿，第14卷，第434号，2020年5月。[4] H. Tachibana, Y. Mizuno, N. Ono, 和 S. Sagayama, 《基于人声抑制和调性转换技术的单声道录音实时音频转音频卡拉OK生成系统》, 信息处理学报，第24卷，第3期，第470–482页，2016年5月。[5] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, D. FitzGerald, 和 B. Pardo, 《音乐主旋律与伴奏分离概述》, *IEEE/ACM* 音频、语音与语言处理汇刊，第26卷，第8期，第1307–1335页，2018年8月。[6] N. Ono, K. Miyamoto, H. Kameoka, J. Le Roux, Y. Uchiyama, E. Tsunoo, T. Nishimoto, 和 S. Sagayama, 谐波与打击乐声音分离及其在音乐信息检索相关任务中的应用。柏林，海德堡：施普林格·柏林·海德堡，2010年，第213–236页。[7] L. Lin, Q. Kong, J. Jiang, 和 G. Xia, 《用于零样本音乐源分离、转录与合成的统一模型》, 载于国际音乐信息检索学会会议（*ISMIR*）论文集，2021年11月，第381–388页。[8] F.-R. Stöter, S. Uhlich, A. Liutkus, 和 Y. Mitsufuji, 《Open-Unmix——音乐源分离的参考实现》, 开放源代码软件杂志，2019年。[9] R. Hennequin, A. Khlif, F. Voituret, 和 M. Moussallam, 《Spleeter：采用预训练模型的快速高效音乐源分离工具》, 开放源代码软件杂志，第5卷，第50期，第2154页，2020年。[10] Y. Luo 和 J. Yu, 《基于频带分割循环神经网络的音乐源分离》, *IEEE/ACM* 音频、语音与语言处理汇刊，第31卷，第1893–1901页，2023年5月。

[11] W.-T. Lu, J.-C. Wang, Q. Kong, 和 Y.-N. Hung, 《基于频带分割绳式变换器的音乐源分离》, 载于*IEEE*声学、语音与信号处理国际会议（*ICASSP*）论文集, 2024年4月。 [12] A. Défossez, 《频谱图与波形混合源分离》, 载于*ISMIR 2021*音乐源分离研讨会论文集, 2021年11月。 [13] K. Drossos, S. I. Mimilakis, D. Serdyuk, G. Schuller, T. Virtanen, 和 Y. Bengio, 《MaD TwinNet：用于单声道声音源分离的双网络掩蔽器-降噪器架构》, 载于*IEEE*神经网络国际联合会议（*IJCNN*）论文集, 2018年7月。 [14] P. Chandna, M. Miron, J. Janer, 和 E. Gómez, 《利用深度卷积神经网络进行单声道音频源分离》, 载于潜变量分析与信号分离, P. Tichavský, M. Babaie-Zadeh, O. J. Michel, 和 N. Thirion-Moreau 编, 沙姆, 2017年, 第258–266页。 [15] S. Rouard, F. Massa, 和 A. Défossez, 《用于音乐源分离的混合变换器》, 载于*IEEE*声学、语音与信号处理国际会议（*ICASSP*）论文集, 2023年6月。 [16] Y. Mitsufuji, G. Fabbro, S. Uhlich, F.-R. Stöter, A. Défossez, M. Kim, W. Choi, C.-Y. Yu, 和 K.-W. Cheuk, 《2021年音乐分离挑战赛》, 《信号处理前沿》, 第1卷, 2022年。 [17] G. Fabbro, S. Uhlich, C.-H. Lai, W. Choi, M. Martínez-Ramírez, W. Liao, I. Gadelha, G. Ramos, E. Hsu, H. Rodrigues, F.-R. Stöter, A. Défossez, Y. Luo, J. Yu, D. Chakraborty, S. Mohanty, R. Solovyev, A. Stempkovskiy, T. Habruseva, N. Goswami, T. Harada, M. Kim, J. H. Lee, Y. Dong, X. Zhang, J. Liu, 和 Y. Mitsufuji, 《2023年声音分离挑战赛音乐分离赛道》, 《国际音乐信息检索学会汇刊》（*TISMIR*）, 2024年4月。 [18] M. Ferrari Dacrema, P. Cremonesi, 和 D. Jannach, 《我们真的取得了很大进展吗？对近期神经推荐方法的令人担忧的分析》, 载于第*13*届*ACM*推荐系统会议（*RecSys*）论文集, 系列：RecSys '19。美国纽约州纽约：计算机协会, 2019年, 第101109页。 [19] A. Belz, S. Agarwal, A. Shimorina, 和 E. Reiter, 《自然语言处理可复现性研究的系统综述》, 载于计算语言学协会欧洲分会第*16*届会议论文集：主会场, P. Merlo, J. Tiedemann, 和 R. Tsarfaty 编, 2021年4月, 第381–393页。 [20] J. Six, F. Bressan, 和 M. Leman, 《 MIR 中的可复现性：对一个高度稳健的音频指纹系统进行复现》, 《国际音乐信息检索学会汇刊》, 2018年9月。 [21] B. McFee, J. W. Kim, M. Cartwright, J. Salamon, R. M. Bittner, 和 J. P. Bello, 《音乐信号处理研究的开源实践：关于透明、可持续和可复现音频研究的建议》, *IEEE* 信号处理杂志, 第36卷, 第1期, 第128–137页, 2019年1月。 [22] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, 和 R. Bittner, 《MUSDB18-HQ——未压缩版本的MUSDB18》, 2019年12月。

[23] [ ] 10 5281 3338373 在线。可用：https://doi.org/ 。/zenodo。D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, 和 D. Dennison, 《机器学习系统中的隐性技术债务》, 载于《神经信息处理系统进展》, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, 和 R. Garnett 编, 第28卷, 2015年。 [24] J. P. Ioannidis, 《是否复现：药物基因组学研究案例》, 《循环：心血管遗传学》, 第6卷, 第4期, 第413–418页, 2013年。 [25] D. H. Johnson, 《复现在野生动物研究中的重要性》, 《野生动物管理杂志》, 第66卷, 第4期, 第919–932页, 2002年。 [26] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, 和 J. Pineau, 《推动系统性报告机器学习的能耗与碳足迹》, 《机器学习研究杂志》（*JMLR*）, 第21卷, 第1期, 2020年1月。[27] E. Strubell, A. Ganesh, 和 A. McCallum, 《现代深度学习研究中的能耗与政策考量》, 载于*AAAI*人工智能会议论文集, 第34卷, 第09期, 2020年4月, 第13 693–13 696页。[28] A. Holzapfel, Anna-Kaisa, 和 K. P. Jääskeläinen, 《绿色MIR？调查ISMIR近期音乐人工智能研究的计算成本》, 载于第*25*届国际音乐信息检索学会会议（*ISMIR*）论文集, 2024年11月, 第371–380页。[29] H. Liu, Q. Kong, 和 J. Liu, 《CWS-PResUNet：采用逐通道子带相位感知ResUNet的音乐源分离》, 载于*ISMIR 2021*音乐源分离研讨会论文集, 2021年11月。

[30] M. Kim、W. Choi、J. Chung、D. Lee 和 S. Jung, 《KUIELab-MDX-Net：一种用于音乐解混的双流神经网络》, 载于 ISMIR 2021音乐源分离研讨会论文集, 2021年11月。[31] Y. Luo 和 R. Gu, 《利用 SIMO 立体声频带拆分 RNN 改进音乐源分离》, 载于 IEEE 声学、语音与信号处理国际会议（ICASSP）论文集, 2024年4月。[32] J. Chen、S. Vekkot 和 P. Shukla, 《基于轻量级深度学习框架的音乐源分离（DTTNET：双路径TFC-TDF UNET）》, 载于 IEEE声学、语音与信号处理国际会议（ICASSP）论文集, 2024年4月。[33] M. Kim、J. H. Lee 和 S. Jung, 《2023声音解混挑战赛音乐解混赛道技术报告：TFC-TDF-UNET V3》, 2023年, arXiv预印本 arXiv:2306.09382。[34] K. N. Watcharasupat、C.-W. Wu、Y. Ding、I. Orife、A. J. Hipple、P. A. Williams、S. Kramer、A. Lerch 和 W. Wolcott, 《用于电影音频源分离的广义频带拆分神经网络》, IEEE开放信号处理期刊, 第5卷, 第73–81页, 2024年。[35] J. Yu、H. Chen、Y. Luo、R. Gu 和 C. Weng, 《利用频带拆分RNN实现高保真语音增强》, 载于 Interspeech 论文集, 2023年8月, 第2483–2487页。[36] Y. Luo、Z. Chen 和 T. Yoshioka, 《双路径RNN：用于时域单通道语音分离的高效长序列建模》, 载于 IEEE声学、语音与信号处理国际会议（ICASSP）论文集, 2020年5月。[37] Y. Luo、Z. Chen、N. Mesgarani 和 T. Yoshioka, 《端到端麦克风排列与数量不变的多通道语音分离》, 载于 IEEE声学、语音与信号处理国际会议（ICASSP）论文集, 2020年, 第6394–6398页。[38] J. Wang, 《基于循环融合膨胀卷积和通道注意力的高效语音分离网络》, 载于 Interspeech 论文集, 2023年8月, 第3699–3703页。[39] A. Pandey 和 D. Wang, 《用于时域语音增强的带自注意力的密集CNN》, IEEE/ACM 音频、语音与语言处理汇刊, 第29卷, 第1270–1279页, 2021年3月。[40] Z.-Q. Wang、S. Cornell、S. Choi、Y. Lee、B.-Y. Kim 和 S. Watanabe, 《TF-GRIDNET：让时频域模型在单耳扬声器分离中重焕活力》, 载于 IEEE声学、语音与信号处理国际会议（ICASSP）论文集, 2023年6月。[41] B. Yan、J. Shi、Y. Tang、H. Inaguma、Y. Peng、S. Dalmia、P. Polák、P. Fernandes、D. Berrebbi、T. Hayashi、X. Zhang、Z. Ni、M. Hira、S. Maiti、J. Pino 和 S. Watanabe, 《ESPnet-ST-v2：多用途口语语言翻译工具包》, 载于计算语言学协会年会论文集, 2023年, 第400–411页。[42] C.-B. Jeon、G. Wichern、F. G. Germain 和 J. Le Roux, 《为何音乐源分离能从刺耳噪声中获益？》, 载于 IEEE声学、语音与信号处理国际会议研讨会（ICASSPW）论文集, 2024年4月, 第873–877页。[43] E. Vincent、R. Gribonval 和 C. Févotte, 《盲音频源分离性能测量》, IEEE语音与音频处理汇刊, 第14卷, 第4期, 第1462–1469页, 2006年7月。[44] F.-R. Stöter、A. Liutkus 和 N. Ito, 《2018年信号分离评测活动》, 载于潜变量分析与信号分离国际会议论文集, 2018年7月。[45] F.-R. Stöter、A. Liutkus、D. Samuel、L. Miner、F. Voituret、pyup.io bot、S. Bot 和 tobeperson, 《sigsep/sigsep-mus-eval：museval 0.4.0》, 2021年2月。 [在线]。获取地址：https://doi.org/10.5281/zenodo.448 6535[46] P. Goyal、P. Dollár、R. B. Girshick、P. Noordhuis、L. Wesolowski、A. Kyrola、A. Tulloch、Y. Jia 和 K. He, 《精准的大批量随机梯度下降：1小时内训练ImageNet》, ArXiv, 第abs/1706.02677卷, 2017年。[47] B. Courty、V. Schmidt、S. Luccioni、Goyal-Kamal、MarionCoutarel、B. Feld、J. Lecourt、LiamConnell、A. Saboni、Inimaz、supatomic、M. Léval、L. Blanche、A. Cruveiller、ouminasara、F. Zhao、A. Joshi、A. Bogroff、H. de Lavoreille、N. Laskaris、E. Abati、D. Blank、Z. Wang、A. Catovic、M. Alencon、M. Stchy、C. Bauer、L. O. N. de Araújo、JPW 和 MinervaBooks, 《mlco2/codecarbon：v2.4.1》, 2024年5月。 [在线]。获取地址：https://doi.org/

[48] 10 5281 11171501 。/zenodo。L. Lannelongue、J. Grealey 和 M. Inouye, 《绿色算法：量化计算的碳足迹》, Advanced Science, 第8卷, 第12期, 第2100707页, 2021年。[49] M. Jay、V. Ostapenco、L. Lefèvre、D. Trystram、A.-C. Orgerie 和 B. Fichel, 《软件功率计的实验比较：聚焦CPU和GPU》, 载于CCGrid 2023——第23届IEEE/ACM集群、云计算与互联网计算国际研讨会。IEEE, 2023年

, 第1–13页。

[50] R. Serizel、S. Cornell 和 N. Turpault，“性能至上？机器听觉中的能耗与性能：对 DCASE 任务4基线的研究”，载于 *IEEE*声学、语音与信号处理国际会议论文集（*ICASSP*），2023年6月。[51] X. Bouthillier、P. Delaunay、M. Bronzi、A. Trofimov、B. Nichyporuk、J. Szeto、N. Mohammadi Sepahvand、E. Raff、K. Madan、V. Voleti、S. Ebrahimi Kahou、V. Michalski、T. Arbel、C. Pal、G. Varoquaux 和 P. Vincent，“ 机器学习基准中的方差核算 ”，载于 机器学习与系统年会论文集（*MLSys*），第3卷，2021年。[52] S. Bethard，“我们需要谈谈随机种子”，2022年，arXiv 预印本 arXiv:2210.13393。[53] J. Heitkaemper、D. Jakobeit、C. Boeddeker、L. Drude 和 R. Haeb-Umbach，“揭开 TasNet 的神秘面纱：一种剖析方法”，载于 *IEEE*声学、语音与信号处理国际会议论文集（*ICASSP*），2020年5月。[54] J. L. Roux、N. Ono 和 S. Sagayama，“短时傅里叶变换频谱图的显式一致性约束及其在相位重建中的应用”，载于 统计与感知音频处理专题研讨会（*SAPA 2008*），2008年，第23–28页。[55] S. Wisdom、J. Hershey、K. Wilson、J. Thorpe、M. Chinen、B. Patton 和 R. A. Saurous，“用于改进深度语音增强的可微一致性约束”，载于 *IEEE*声学、语音与信号处理国际会议论文集（*ICASSP*），2019年5月。[56] C. Douwes、G. Bindi、A. Caillon、P. Esling 和 J.-P. Briot，“质量是否足够？在大规模神经音频合成模型评估中纳入能耗”，载于 *IEEE*声学、语音与信号处理国际会议论文集（*ICASSP*），2023年6月。[57] Z. Rafii、A. Liutkus、F.-R. Stöter、S. I. Mimilakis 和 R. Bittner，“用于音乐分离的 MUSDB18 语料库”，2017年12月。 [在线]。网址：https://doi.org/ 10 5281 1117372 。/ zenodo。[58] Q. Kong、Y. Cao、H. Liu、K. Choi 和 Y. Wang，“利用深度 ResUNet 解耦音乐源分离中的幅度与相位估计”，载于 第*22*届国际音乐信息检索学会会议论文集（*ISMIR*），2021年11月，第342–349页。[59] R. Sawata、S. Uhlich、S. Takahashi 和 Y. Mitsufuji，“人人为我，我为人人：通过连接网络改进音乐分离”，载于 *IEEE*声学、语音与信号处理国际会议论文集（*ICASSP*），2021年6月，第51–55页。[60] Y. M. Naoya Takahashi，“D3Net：用于音乐源分离的密集连接多膨胀率 DenseNet”，2023年，arXiv 预印本 arXiv:2010.01733。