

FCPE: 一种快速上下文依赖的音高估计模型

Yuxin Luo^{1,†}, Ruoyi Zhang^{1,†}, Lu-Chuan Liu², Tianyu Li¹, Hangyu Liu^{1,*}

¹Fish Audio, Santa Clara, CA, USA

²University of Science and Technology of China, Hefei, Anhui, China

ABSTRACT

音高估计 (PE) 在单声道音频中对于 MIDI 转录和歌声转换 (SVC) 至关重要, 但现有方法在噪声条件下性能显著下降。本文提出了 FCPE, 一种快速基于上下文的音高估计模型, 该模型采用具有深度可分离卷积的 Lynx-Net 架构, 在保持低计算成本和鲁棒的噪声容忍度的同时有效捕捉梅尔频谱图特征。实验表明, 我们的方法在 MIR-1K 数据集上实现了 96.79% 的原始音高准确率 (RPA), 与最先进的方法相当。实时因子 (RTF) 在单个 RTX 4090 GPU 上的值为 0.0062, 在效率方面显著优于现有算法。代码可在 <https://github.com/CNChTu/FCPE> 获取。

Index Terms— 音高估计, 快速推理, 深度学习

1. 介绍

音高估计 (PE) 或基频 (f0) 估算是诸如 MIDI 转录和歌唱声音转换 (SVC) 等任务中的关键部分。声乐音高估算是广泛用于工业生产的一个案例。在近期深度学习模型出现之前, PE 领域主要由经典信号处理技术主导。这些方法可以分为三类。

- **时域方法**: 它们直接作用于信号波形以识别其周期结构。最著名的一种是自相关函数 (ACF) 方法 [1]。
- **频域方法**: 这些方法将信号转换到频率域以分析其谐波结构。基于倒谱的音高确定算法 [2] 是一个经典示例, 旨在分离声门源激励和声道滤波器以揭示基频。
- **混合方法**: 为了实现更大的鲁棒性, 这些方法结合了两个领域的技术。例如, 开创性的 YIN 算法 [3] 通过加入几个误差减少步骤来增强自相关函数, 而 YAAPT [4] 则明确地将频谱谐波信息与时域分析结合起来以提高跟踪精度。

上述方法代表了 PE 的传统范式, 虽然取得了显著的成功, 但仍难以应对噪声环境、多声部来源等问题。

随着深度学习 (包括软件和硬件) 的发展, 相关工作极大地提高了 PE 任务的结果。开创性的模型如 CREPE [5]、DeepF0 [6] 和 HARMOF0 [7] 利用了卷积神经网络 (CNN) 或循环神经网络 (RNN), 为准确性和鲁棒性建立了新的标杆。最近, RMVPE [8] 的引入通过适应像 U-Net 这样的新模型达到了前所未有的性能, 标志着另一个重要的里程碑。然而, 当前的深度学习方法需要大量的计算并引入了延迟, 限制了它们在实时应用中的使用。特别是对于 RMVPE 而言, 其高计算需求是由于其复杂的架构直接导致的结果。

为了解决这些问题, 我们提出了快速上下文基音估计 (FCPE), 这是一个在不牺牲准确性的情况下实现高效的新模型。FCPE 采用 Lynx-Net 主干网络 [9], 该网络利用深度可分离卷积从梅尔频谱图中高效地提取特征, 并提供足够的上下文覆盖以建模帧间的时序关系。在这篇论文中, 我们展示了这种架构选择与精心设计的训练策略相结合, 共同使我们的模型达到最先进的性能。

[†]These authors contributed equally to this work.

*Corresponding Author.

2. FCPE

2.1. 总体架构

我们将模型输入定义为 $X_{T \times F}$, 其中 X 表示对数梅尔频谱图, T 是梅尔帧的数量, 而 F 是对数间隔的频率 bin 数量。我们的任务可以表述为 $F: X_{T \times 128} \rightarrow Y_{T \times 360}$, Y 是预测的音高矩阵。整体结构如图 1 所示, 可以分解为三个阶段:

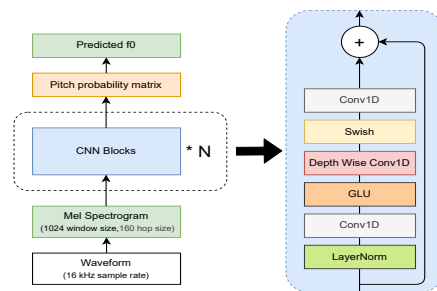


Fig. 1: FCPE 的整体架构。

• **输入表示**: 如图 1 所示, 该过程从原始音频波形开始, 首先将其转换为对数梅尔频谱图。这种频谱表示随后通过初始嵌入块。这个由浅层一维卷积层组成的模块将输入特征映射到一个适合主干的高维向量序列。可选地, 在输入下一阶段之前, 可以为此序列添加可学习的谐波嵌入以明确增强谐波特征。

• **Lynx-Net 主干网络**: FCPE 的核心是由轻量级 Lynx-Net 层 (图 1 中的 CNN 块) 堆叠而成, 旨在高效地建模时间上下文。如图 1 所示, 每个块都使用了一种受 Conformer 启发的 [10] 结构。其关键组件是一个深度可分离 Conv1D 层, 可以高效捕捉局部模式。逐点卷积管理通道维度, 并且残差连接有助于训练更深的网络。

• **输出阶段**: 经过 Lynx-Net 层处理后, 细化的特征序列通过最终的线性层投影以生成音高概率矩阵 $Y_{T \times 360}$ 。该矩阵表示每个时间帧内每一分 bin 的概率。为了得到最终的 f0, 我们不仅选取具有最高概率的 bin。相反, 我们在解码步骤中应用了一个局部 argmax 函数 [5, 8]。此函数计算峰值概率 bin 周围音高的加权平均值, 从而提供比简单 argmax 操作更精确和稳健的 f0 估计, 生成最终预测的 f0。

2.2. 损失函数和解码策略

遵循之前的工作 [5, 8], 我们将音高估计表述为一个在 360 个离散音高区间上的分类问题。每个区间对应一个相对于 $f_{\text{ref}} = 10$ 赫兹定义的具体音高值:

$$c(f) = 1200 \cdot \log_2 \frac{f}{f_{\text{ref}}} \quad (1)$$

360 个音高值表示为 $c_1, c_2, \dots, c_{360}$ ，并选择覆盖从 C1 到 B7 的六个八度，这两个音分别对应 32.70 Hz 和 1975.5 Hz，其间隔为 20 音分。这个对数音阶每半音有 100 音分，从而实现了精细的音高分辨率。

模型输出一个概率向量 $\hat{y}$ 在这些区间上。我们使用 BCE 损失进行训练：

$$\mathcal{L}(y, \hat{y}) = - \sum_{i=1}^{360} (y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)), \quad (2)$$

其中目标 y 定义与 CREPE 论文中完全相同。

在推理过程中，我们采用局部加权平均解码机制 [8] 来获得最终的音高估计。如果置信度分数超过阈值，则以峰值概率周围的加权平均值计算音高估计 $\hat{c}$ （单位为分）。

$$\hat{c} = \sum_{i=m-4}^{m+4} (\hat{y}_i c_i) / \sum_{i=m-4}^{m+4} \hat{y}_i, \quad m = \arg \max_i \hat{y}_i \quad (3)$$

$$\hat{f} = \begin{cases} f_{\text{ref}} \times 2^{\hat{c}/1200} & \text{if } \max_i \hat{y}_i \geq 0.05 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

2.3. 训练详情

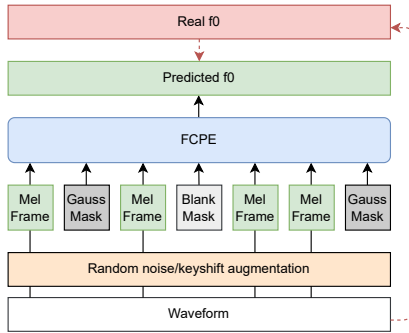


Fig. 2: 训练策略的详情。

为了建立客观精确的地面真实数据并消除人工标注带来的主观错误，我们使用可微数字信号处理 (DDSP) [11] 方法重新综合了 M4Singer[12] 和 VCTK[13] 数据集，并将它们用于训练所有模型变体。然后，我们采用几种数据增强策略。首先，我们将随机键移应用到原始音频波形上以增加音高多样性。接下来，为了提高模型的噪声鲁棒性，我们在信号上叠加了各种类型的噪声（白噪声、彩色噪声、现实世界噪声）。最后，我们将随机掩码应用于增强后的梅尔频谱图，如图 2 所示，这涉及空白或高斯掩码。这种方法迫使模型从周围的上下文信息中推断音高，而不是依赖单帧的孤立特征。我们在实验中实证验证了这些技术的有效性，结果显示在噪声条件下模型鲁棒性有了显著提高。

3. 实验

为了比较 FCPE 与其他模型，并量化训练策略对系统的影响，我们进行了一系列实验。以下部分展示了对比研究和消融研究的结果。

3.1. 比较研究

3.1.1. 准确率

为了展示我们模型的有效性和鲁棒性，我们对其与五种领先的音高估计算法进行了比较研究：RMVPE、CREPE、PESTO[14]、PM[15] 和 Harvest[16]。性能评估在 MIR-1K[17, 18]、Vocadito[19] 和 TONAS[20] 数据集上进行，这些数据集合包括各种具有挑战性的条件下的音频，如干净的音频和受到彩色噪声及来自 CHiME 数据集 [21] 的真实世界环境噪声污染的音频，在多个信噪比 (SNRs) 下进行了测试。此外，为了减少手动标注的误差，我们还在评估中使用了另一个 DDSP 重新合成的数据集。具体来说，它是从 THCHS30 [22] 数据集的测试集中重新合成的，我们将这个数据集称为 THCHS30-Synth。在这次比较中，所有基于深度学习的模型都使用它们公开可用的预训练版本进行评估 [23]。需要注意的是，由于 MIR-1K 数据集在 RMVPE、CREPE 和 PESTO 的训练中被使用过，因此预计这些模型在这个特定数据集上的表现会更优。这四个数据集包含各种声乐数据，涵盖了从歌唱到语音的广泛范围。

原始音高准确性 (RPA) 在表 1 中给出。所有评估进行了五次，平均值在表格中报告。如表所示，我们的模型在多个数据集上表现出显著的性能，在存在严重噪声干扰的情况下，仍展现了出色的鲁棒性。结果清楚地表明，FCPE 的准确性与最先进的稳健模型 RMVPE 具有高度竞争力。这种一致、高水平的表现验证了我们的方法，证明 FCPE 为广泛的场景提供了卓越的鲁棒性和准确性，使其成为音高估计的强大且高效的工具。

另一方面，我们的模型并没有像 RMVPE 或 CREPE 那样拥有众多参数，但我们依然达到了与之相当甚至超越其性能的水平，这充分展示了我们模型架构和训练策略的有效性。

3.1.2. 实时因子

除了准确性，计算效率是实时歌声转换等实际应用中的关键因素。为了评估 FCPE 的推理速度，我们测量其实时因子 (RTF)。RTF 是衡量音频处理模型效率的标准指标，定义为处理音频流所需的时间除以音频本身的时长。一个 RTF 远低于 1.0 的模型被认为能够进行实时处理。RTF 的计算方法如下：

$$\text{RTF} = \frac{T_{\text{process}}}{T_{\text{audio}}} \quad (5)$$

其中 T_{process} 是模型推理音频片段音高所需的时间，该音频片段的时长为 T_{audio} 。

我们还计算了处理一秒音频所需的浮点运算次数 (FLOPS)，较低的 FLOPS 需求也意味着较低的推理成本。

结果，如表 2 所示，明确地证明了 FCPE 的计算效率。我们的模型实现了 0.0062 的 RTF，这比 RMVPE 快大约 5.3x 倍，即使其参数远少于我们模型，也比 PESTO 快 2.6x 倍，并且比广泛使用的 CREPE 模型惊人地快 77x 倍。

显著的速度优势直接归因于 FCPE 的轻量级架构，该架构大量依赖高效的深度可分离卷积。这不仅证实了 FCPE 适用于实时应用，如实时语音转换和音乐信息检索，还使其成为大规模模批处理以及在资源受限的边缘设备上部署的理想候选者。

3.2. 消融研究

为了研究每种训练策略对模型性能的单独贡献，我们进行了一系列消融研究。我们的模型是通过结合三种关键的数据增强技术进行训练的：**噪声增强**、**频谱图掩码**和**随机密钥移位**。然后，我们在训练过程中系统地移除这些组件中的每一个，从而创建模型的不同变体。

所有模型变体都在第 2.3 节提到的同一重新合成数据集上进行训练，并在 MIR-1K 数据集上进行评估。为了严格评估鲁棒性，我们在一系列合成噪声条件下进行了评估，其中我们改变了噪声颜色（由 β 参数控制）和信噪比 (SNR 以 dB 为单位)。

Table 1: 六种算法的性能比较。未列出参数数量的是传统的信号处理算法。

算法 (Parameters)	MIR-1K 数据集				THCHS30-合成数据集				Vocadito 数据集				TONAS 数据集			
	RPA (%) 上升				RPA (%) 上升				RPA (%) ↑				RPA (%) 上升			
	清理	20 分贝	0 分贝	-20 分贝	清洁	20 分贝	0 分贝	-20 分贝	清洁	20 分贝	0 分贝	-20 分贝	清洁	20 分贝	0 分贝	-20 分贝
白噪声 ($\beta=0$)																
FCPE (10.64M)	96.79	97.06	97.09	29.75	97.56	96.48	81.44	12.17	95.80	95.82	93.43	21.60	96.03	95.86	64.56	
RMVPE (90.42M)	97.77	97.57	97.39	43.63	96.37	95.79	84.12	3.98	95.83	95.81	95.47	28.80	95.64	95.66	81.49	
CREPE (22.24M)	97.90	97.89	94.07	1.09	89.67	88.81	65.78	0.16	97.35	97.35	93.08	0.56	95.20	95.20	79.69	
PESTO (0.13M)	98.47	98.38	95.48	17.78	89.72	86.47	60.59	7.01	95.50	95.24	89.18	13.20	91.30	89.31	64.59	
PM	96.06	95.63	20.90	0.00	77.57	77.52	14.14	0.00	91.75	91.62	19.27	0.00	89.96	88.83	10.81	
Harvest	95.11	95.03	62.02	0.17	87.93	87.06	40.70	0.05	94.17	93.88	72.19	0.17	93.98	93.00	11.27	
粉红噪声 ($\beta=1$)																
FCPE (10.64M)	96.79	97.12	95.97	18.04	97.56	96.30	76.42	8.40	95.80	95.88	92.28	15.65	96.03	95.81	61.74	
RMVPE (90.42M)	97.77	97.59	96.61	13.66	96.37	95.34	77.46	2.83	95.83	95.84	94.82	9.15	95.64	95.72	77.29	
CREPE (22.24M)	97.90	97.79	91.72	2.54	89.67	87.61	59.03	0.62	97.35	97.27	90.92	1.91	95.20	94.66	73.05	
PESTO (0.13M)	98.47	98.39	92.62	10.24	89.72	86.34	61.97	6.68	95.50	95.16	88.09	9.90	91.30	90.05	67.42	
PM	96.06	96.02	49.84	0.00	77.57	77.89	26.66	0.00	91.75	91.78	52.99	0.00	89.96	89.61	42.76	
Harvest	95.11	94.98	51.06	0.06	87.93	85.89	27.81	0.04	94.17	93.68	62.04	0.10	93.98	92.52	9.25	
真实世界噪声 (CHiME)																
FCPE (10.64M)	96.78	96.95	90.59	35.83	97.56	96.47	80.51	23.87	95.80	95.87	91.17	38.03	96.03	95.88	76.83	
RMVPE (90.42M)	97.77	97.89	95.13	41.53	96.37	95.56	84.37	30.81	95.83	95.85	93.71	43.72	95.64	95.62	86.56	
CREPE (22.24M)	97.90	97.70	91.43	30.20	89.67	87.64	66.37	13.66	97.35	97.24	91.16	31.62	95.20	94.54	76.44	
PESTO (0.13M)	98.47	98.34	91.14	35.84	89.72	88.41	71.42	23.74	95.50	95.38	89.24	38.43	91.30	90.33	73.32	
PM	96.06	95.39	62.91	4.15	77.57	76.70	44.86	3.41	91.75	91.36	65.89	2.96	89.96	89.27	63.02	
Harvest	95.11	94.87	77.59	12.85	87.93	86.24	58.17	8.35	94.17	93.81	80.36	15.89	93.98	92.15	56.58	

Table 2: 实时因子比较和不同音高估计模型推理一秒钟所需的FLOPS。测试是在单个 RTX 4090 GPU 上运行的。

模型	准备就绪	每秒浮点运算次数
FCPE (我们的)	0.0062	1.06 千亿次浮点运算/秒
PESTO (MIR-1K_g7)	0.0164	2.82 GFLOPS
RMVPE	0.0329	4.91 GFLOPS
CREPE	0.4775	141 GFLOPS

我们使用两个指标进行评估：RPA 和 RCA。详细结果如下表 1 所示。实验重复五次，并将平均值报告在表格中（与比较研究相同）。

表 3 中的结果清晰地展示了每个组件的贡献。首先，很明显，在没有训练**噪声增强**的模型在噪声条件下准确性大幅下降。这一点在极端的-20 分贝白噪声 ($\beta=0$) 情况下最为明显，RPA 从基线的 29.75%骤降至仅 6.19%。这种明显的退化明确表明，噪声增强是提高模型鲁棒性的最关键组件。

其次，**频谱图掩码**的重要性也得到了明确展示，特别是在对抗结构噪声方面。在其移除后，模型在 -20 dB 粉红噪声 ($\beta=1$) 上的 RPA 下降到仅为 4.81%。值得注意的是，这甚至比没有噪声增强的模型 (6.00%) 更差，这强烈支持了我们的核心假设：掩蔽是促使 FCPE 利用其长上下文架构的关键机制。这种从时间上下文中寻找可靠音高线索的学习技能对于克服极端噪声至关重要。

最后，移除**键移位**显示了模型精度与泛化之间的有趣且重

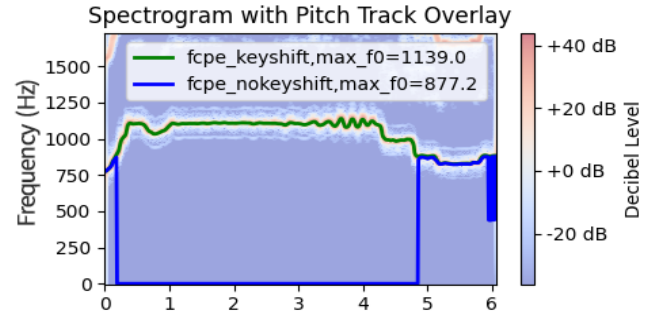


Fig. 3: 模型在有无调性变换情况下的音域。蓝色线条错误地预测了中间部分没有基频，而绿色线条（带有调性变换）则保持连续的音高跟踪。

要的权衡。乍一看，没有这种增强的模型似乎更优越，在所有变体上的干净数据中实现了最高准确率。然而，我们的测试集 (MIR-1K) 无法覆盖更加普遍和现实的情况，并且我们合成训练数据中存在的音高多样性不足需要通过键移位来提升真实场景下的泛化能力。如图 3 所示，没有键移位增强训练的模型错误地将一个高音人声段识别为无声（无 f_0 ），而经过键移位增强训练的模型则正确检测到了同一段落中的 f_0 。进一步的定量分析显示，带有键移位增强的模型的音域要 29.8% 更宽（1139 赫兹 对比 877.2 赫兹）。

总而言之，消融研究的发现可以总结如下：

Table 3: 不同训练策略在 MIR-1K 数据集上的消融结果按噪声类型组织。(↑: 越高越好)。

策略	白噪声 ($\beta = 0$)				粉红噪声 ($\beta = 1$)				布朗噪声 ($\beta = 2$)				紫噪声 ($\beta = -1$)			
	清洁	20 分贝	0 分贝	-20 分贝	清洁	20 分贝	0 分贝	-20 分贝	清洁	20 分贝	0 分贝	-20 分贝	清洁	20 分贝	0 分贝	-20 分贝
完整策略	96.79	97.06	97.09	29.75	96.79	97.12	95.97	18.04	96.79	96.91	96.99	96.31	96.79	96.88	97.09	41.8
无噪声增强。	96.69	96.88	94.35	6.19	96.69	96.99	91.75	6.00	96.69	96.87	97.03	92.46	96.69	96.79	95.71	18.1
无 Spec. 掩码。	96.58	96.87	96.97	16.72	96.58	96.93	95.64	4.81	96.58	96.66	96.75	96.00	96.58	96.72	97.03	55.4
无密钥移位。	96.94	97.18	97.20	32.77	96.94	97.23	96.02	9.14	96.94	97.05	97.14	96.61	96.94	97.08	97.37	66.4

• **噪声数据对模型的性能至关重要**：包含所有增强技术的 FCPE 模型在所有指标上都表现出了强大且平衡的性能。这表明我们提出的训练策略对于生成一个稳健和准确的音高估计模型是有效的。

• **频谱图掩码增强鲁棒性**：带有和不带有频谱掩蔽的实验表明，由于我们的模型具有长上下文架构，能够有效捕捉前后帧之间的时间关系，并且在极端噪声条件下表现出鲁棒性。

• **关键移位扩展音域**：初看之下，移除键位移动（无键位移动）在某些准确性指标上略有提升。然而，鉴于我们的训练数据音高范围有限，这种增强是必要的权衡。它使模型能够覆盖更广泛的音乐范围，并提高泛化能力，这对于实际应用至关重要。

4. 结论

在本文中，我们提出了 FCPE，一个达到最新技术水平的快速基于上下文的音高估计模型。我们还介绍了几种训练策略来增强模型性能，这些策略通过消融研究被证明是有效的。此外，我们成功地利用 DDSP 重合成数据解决了数据稀缺问题，并发现即使重合成的数据也表现出优秀的泛化能力。我们的未来工作将重点应用于 FCPE 在实时 SVC 及其他相关任务中的应用。

5. REFERENCES

- [1] L. Rabiner, M. Cheng, et al., “A comparative performance study of several pitch detection algorithms,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 24, no. 5, pp. 399–418, 1976.
- [2] A. Michael Noll, “Cepstrum Pitch Determination,” The Journal of the Acoustical Society of America, vol. 41, no. 2, pp. 293 – 309, Feb. 1967.
- [3] Alain De Cheveigné et al., “YIN, a fundamental frequency estimator for speech and music,” The Journal of the Acoustical Society of America, vol. 111, no. 4, pp. 1917–1930, 2002.
- [4] Stephen A Zahorian et al., “YAAPT: A robust algorithm for pitch tracking,” The Journal of the Acoustical Society of America, vol. 129, no. 4, pp. 2626–2626, 2011.
- [5] Jong Wook Kim, Justin Salamon, et al., “CREPE: A convolutional representation for pitch estimation,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 5739–5743.
- [6] Sangeet Singh, Ruoqi Wang, et al., “Deepf0: End-to-end fundamental frequency estimation for music and speech signals,” in 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 61–65.
- [7] Weixing Wei, Peilin Li, et al., “Harmof0: Logarithmic scale dilated convolution for pitch estimation,” in 2022 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2022, pp. 1–6.
- [8] Haojie Wei, Xueke Cao, et al., “RMVPE: A Robust Model for Vocal Pitch Estimation in Polyphonic Music,” arXiv preprint arXiv:2306.15412, 2023.
- [9] yxlllc, “LynxNet implementation in DDSP-SVC,” <https://github.com/yxlllc/DDSP-SVC/blob/6.1/reflow/lynxnet.py>, GitHub repository, branch 6.1, file reflow/lynxnet.py.
- [10] Anmol Gulati, James Qin, et al., “Conformer: Convolution-augmented Transformer for Speech Recognition,” 2020.
- [11] Jesse Engel, Lamtharn (Hanoi) Hantrakul, et al., “DDSP: Differentiable Digital Signal Processing,” in International Conference on Learning Representations, 2020.
- [12] Lichao Zhang, Ruiqi Li, et al., “M4Singer: A Multi-Style, Multi-Singer and Musical Score Provided Mandarin Singing Corpus,” in Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2022.
- [13] Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonal, “CSTR VCTK Corpus: English Multi-speaker Corpus for CSTR Voice Cloning Toolkit (version 0.92),” 2019.
- [14] Alain Riou, Stefan Lattner, Gaëtan Hadjeres, and Geoffroy Peeters, “PESTO: Pitch Estimation with Self-supervised Transposition-equivariant Objective,” in Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023. 2023, International Society for Music Information Retrieval.
- [15] P. Praat Boersma, “Praat, a system for doing phonetics by computer,” glot international, 2000.
- [16] Masanori Morise, “Harvest: A High-Performance Fundamental Frequency Estimator from Speech Signals,” in Interspeech 2017, 2017, pp. 2321–2325.

- [17] Chao-Ling Hsu and Jyh-Shing Roger Jang, “On the improvement of singing voice separation for monaural recordings using the MIR-1K dataset,” *IEEE transactions on audio, speech, and language processing*, vol. 18, no. 2, pp. 310–319, 2009.
- [18] Colin Raffel, Brian McFee, et al., “mir_eval: A Transparent Implementation of Common MIR Metrics,” in *Proceedings of the 15th International Conference on Music Information Retrieval*, 2014.
- [19] Rachel Bittner, Katherine Pasalo, et al., “voadito: A dataset of solo vocals with f0, note, and lyric Annotations ,” Oct. 2021.
- [20] COFLA (COmputational analysis of FLAmenco music) team, “TONAS: a dataset of flamenco a cappella sung melodies with corresponding manual transcriptions ,” Mar. 2013.
- [21] Peter Foster, Siddharth Sigtia, et al., “Chime-home: A dataset for sound source recognition in a domestic environment,” in *2015 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2015, pp. 1–5.
- [22] Zhiyong Zhang Dong Wang, Xuewei Zhang, “THCHS-30 : A Free Chinese Speech Corpus,” 2015.
- [23] “Algorithm implementations,” The public model checkpoints and inference codes for algorithms CREPE, RMVPE, and PESTO are from <https://github.com/maxrmorrison/torchcrepe>, <https://github.com/yxlllc/RMVPE>, and <https://github.com/SonyCSLParis/pesto>, respectively.