LJ-SPOOF：用于音频反欺骗与合成源追踪的生成多样化语料库

Surya Subramani、*Hashim Ali*、*Hafiz Malik*

电气与计算机工程系，密歇根大学，4901
Evergreen Rd, Dearborn, MI 48128,USA

arXiv:2601.07958v1 [cs.SD] 12 Jan 2026

摘要

说话人特定反欺骗与合成源追踪是音频反欺骗领域的核心挑战。由于缺乏系统性改变模型架构、合成流程和生成参数的数据集，相关研究进展受到阻碍。为弥补这一空缺，我们推出**LJ-Spoof**，这是一个说话人特定且生成多样化的语料库，系统性改变韵律、声码器、生成超参数、真实语音提示源、训练方案和神经后处理。该语料库涵盖一位说话人（包括录音室级质量的录音）、30 个文本转语音家族、500 个生成变体子集、10 个真实语音神经处理变体，以及超过 300 万条话语。这种密集变化的设计有助于实现稳健的说话人条件反欺骗和细粒度的合成源追踪。我们还将该数据集定位为实用的参考训练资源，以及用于反欺骗和源追踪的基准评测套件。

索引词— 音频反欺骗、说话人验证、深度伪造、来源追踪、合成语音

1. 引言

大语言模型时代的语音合成已经达到足以经常欺骗人类听众的自然度。自监督学习和离散语音标记化技术的发展，加上能够捕捉非语义线索（说话人身份、风格、韵律、情感）的强大声学编码器，进一步缩小了合成语音与真实语音之间的差距。尽管这些进步有利于语音应用，但也降低了被滥用的门槛[1]：开源系统仅凭 5–10 秒的参考音频即可克隆声音，并在数分钟内生成具有说服力的语音，从而助长大规模欺诈和舆论操纵。

除广泛的跨说话人检测外，保护个体声音也至关重要。对于普通用户，克隆声音可能被用来伤害其家人或同事；对于公众人物，克隆声音则可能影响叙事走向并左右公众舆论。因此，针对特定说话人的反欺骗研究是一个互补且亟需推进的研究方向。

综述（例如 [2]）概述了神经文本转语音的发展现状，而近期关于可控文本转语音的研究 [3] 表明，通过调整参数（例如温度、求解器步数、时长/速度）和配置，现代生成器能够根据同一提示生成大量变体。由此引出一个核心问题：当前的反欺骗系统能否抵御这些生成变体？它们能否区分语义或韵律经过操纵的伪造语音、经过神经后处理或重新合成的真实语音，以及未经处理的真实语音？要回答这些问题，就需要仔细审视现有的反欺骗数据集——它们是模型开发与评估的起点；这也促使我们开展本文所述的分析并设计相应的数据集。

2. 音频反欺骗数据集及其不足

现有语料库涵盖单说话人 TTS [4, 5]，多说话人 TTS/VC [6, 7, 8, 9]，多语言 [10]，以扩散为重点的语料库 [11, 12]，以及基于编解码器的语料库 [13]。ASVspoof 系列不断提升攻击的真实性，但每位说话人的真实语音和 伪造语音材料仍然不足，无法支持严谨的单说话人分析。**CodecFake-Omni** [13] 是规模最大的神经编解码器数据集（基于 31 个编解码器家族训练；评估 17 个 CoSG 模型）。以扩散为中心的语料库则形成了补充：**DFADD** [12]基于五种扩散/流匹配生成器（英语/中文）进行训练，其等错误率优于仅使用 ASVspoof 的基线；**DiffSSD**[11]汇集了十种扩散系统，揭示了检测器盲点，并表明重新训练可以恢复性能。**Diffuse orConfuse** [4]（LJSpeech）提供四种扩散合成语音，以及对非扩散伪造语音进行扩散再声码化的样本，重点在于语料筛选与分析，而非完整的反制训练。

单说话人语料包括**WaveFake** [5]（通过神经声码器对真实语音进行重新声码器编码）和**Diffuse or Confuse**（扩散声学 + 重新声码化）。**MLAADv5** [10]汇集了来自 38 种语言、基于 42 种架构的 91 个子集，之后归为 27 个文本转语音家族；伪造语音并非针对特定说话人，重点在于多语言覆盖。**STOPA** [14]旨在开放世界场景下追踪来源，包含 13 种合成器（基于 VCTK）的 70 万个样本。

数据集	Type	TTS	语音变化	生成变化	语音重建	重新编码	前向变化	子集	合成
TIMIT-TTS	多语言	12	✗	✗	✗	✗	✗	12	79k
FoR	多语言	7	✗	✗	✗	✗	✗	7	87k
真实环境	多语言	–	–	–	–	–	–	–	11k
ASVspoof2019	多语言	6	✓	✗	✗	✗	✗	17	108k
ASVspoof2021	多语言	6	✓	✗	✗	✓	✗	17	148k
ASVspoof 5	多语言	15	✓	✗	✗	✓	✗	32	1,211k
CodecFake-Omni	多语言	17	✗	✗	✗	✓†	✗	17	–
DFADD	多语言	5	✗	✗	✗	✗	✗	5	163k
DiffSSD	多语言	10	✗	✗	✗	✗	✗	10	70k
WaveFake	单	0	✗	✗	✗	✗	✗	10	118k
扩散/混淆	单	8	✗	✗	✓	✗	✗	12	183k
STOPA	多语言	6	✓	✗	✗	✗	✓	13	699k
MLAAD	多语言	27	✗	✗	✗	✗	✓	91	175k
著名人物	多语言	10	✗	✗	✗	✗	✓	10	265k
LJ-Spoof	**单**	**30**	✓	✓	✓	✓	✓	**500**	**2,800k**

表 1。反欺骗数据集及合成特征。缩写：VocVar = 基于声码器；GenVar =生成参数；ReVoc =重新声码化；ReCod = 编解码器重合成；FWVar = 基于框架。 †仅真实语音。

总体而言，这些数据集分别拓展了特定维度——多说话人、单说话人、扩散、编解码器和多语言——但没有任何一个提供受控且全面的单说话人资源，能够在固定外部因素的同时，系统地改变架构、生成设置、重新声码化和编解码器重合成。表 1 清楚地展示了现有数据集与我们新建的 LJ-Spoof 之间的差异。

我们的研究推出了全新的数据集**LJ-Spoof**，其特点包括：

- **33 文本转语音家族：** 11 种单说话人文本转语音（SS-TTS）、2 种可微调的多说话人文本转语音（FT-TTS）和 20 种零样本（ZS-TTS），涵盖最先进系统，使该语料库成为独特聚焦于文本转语音合成欺骗语音的语料库。这些家族涵盖了从基础循环神经网络到基于大语言模型的架构等所有主流范式。

- **涵盖的训练方案：** 11 个在目标说话人的真实语音音频上预训练的 SS-TTS 家族、2 个微调 TTS 家族，以及 20 个零样本文本转语音家族，共同隔离单说话人、微调和零样本合成的影响。

- **500变体子集：** 通过训练方案、合成输入、生成参数、声学模型与声码器配对以及神经后处理进行系统性变化，产生了以往研究中鲜少探索的丰富且密集的配置。

- **可复现协议：** 一份详细且带版本记录的协议文件记录了架构选择、配置设置和神经后处理步骤，使研究社区能够开展严谨、细粒度的单说话人反欺骗分析。

表 1 清楚地表明，LJ-Spoof 是一个仅包含单说话人的数据集，规模庞大且在合成方面具有多样性，涵盖基于声码器的变化、生成参数变化、重新声码化、Re-Codec（编解码器重合成）以及基于框架的变化。

3. LJ-SPOOF

本节总结了*LJ-Spoof*数据集的构建过程、背后的设计选择以及分析目标。

3.1. 真实语音和默认子集

我们的目标是在单说话人、录音室级质量的设置下，隔离合成器和参数的影响。LJSpeech [15]包含 13,100 条话语（平均 6.57

, s) 来自一名单一叙述者（共 ≈24 小时），支持 SS-TTS 训练、微调文本转语音（FT-TTS）微调，以及零样本文本转语音（ZS-TTS）参考提示。我们将这 13k 条话语视为默认的 真实语音集。使用该默认集，我们在 30 个文本转语音家族中生成了 34 个默认合成子集，每个子集均采用其已发布或建议的配置。

3.2. 变体子集

除这 35 个默认子集之外，我们还将基于训练类别、声学模型-声码器配对、参考话语和文本输入、生成参数以及神经后处理变化，创建超过 500 个合成子集，如表2 所述。我们仅使用官方代码库、Coqui TTS 和 Nvidia Nemo 来实现默认子集和变体子集。

3.2.1. 训练类别变体

训练类别指文本转语音的训练方法。具体而言，它可以是针对单个说话人进行预训练的模型（SS-TTS）[16], 、在多说话人数据集上预训练并能够针对单个说话人微调其说话人编码器的模型（FT-TTS）[17], 以及能够进行零样本推理、在大型多说话人数据集上预训练的模型（ZS-TTS）[18]。大多数文本转语音家族都是针对特定训练类别进行训练的，但 [19]包含全部三种训练类别的权重，而 [17] 包含 FT-TTS 和 ZS-TTS。

3.2.2. 声学-声码器变体

大多数 SS-TTS 家族使用在 LJSpeech 上预训练的声学模型；LJSpeech 实际上是 SS-TTS 实现中默认使用的语料库。这些声学模型通常与神经声码器级联， [20] 这些声码器也提供在 LJSpeech 上预训练的开源权重。这种由独立预训练组件构成的级联设计，支持灵活搭配声学模型和声码器模块，前提是声码器支持声学模型输出的梅尔频谱图配置（例如梅尔频带数、频率范围）。由于声学模型的梅尔输出规格与声码器所需的梅尔输入配置不匹配，少数搭配无法使用。

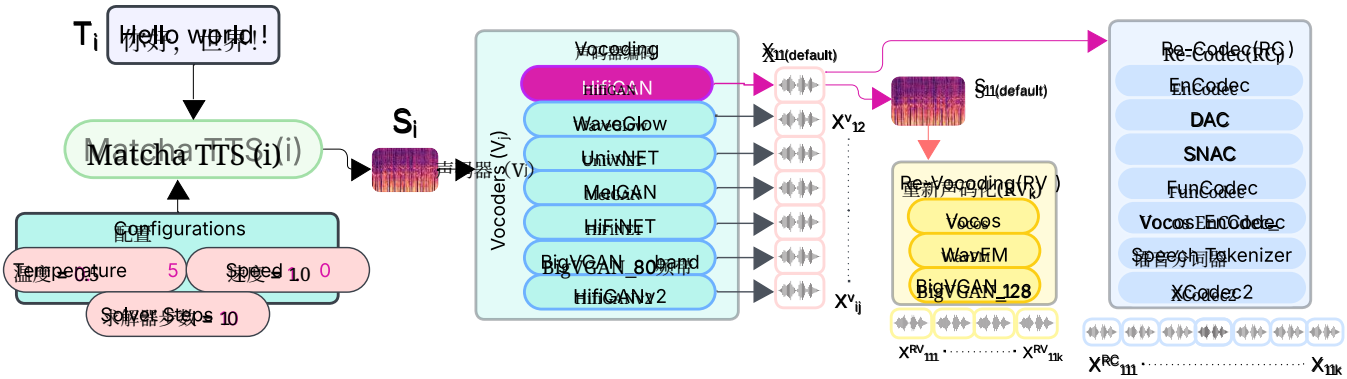

图 1。Matcha-TTS 的数据集生成工作流。使用默认生成参数（采样温度 $\tau = 0.5$、峰值速率 $R = 1.0$、常微分方程求解器步数 $D = 10$），从文本 T合成梅尔频谱图 S。随后，多个声码器（蓝色）会对其进行解码，其中包括默认的 HiFiGAN（紫色），从而生成多种波形子集。最后，通过重新声码化（黄色）和重新编解码（蓝色）对默认波形进行后处理。

3.2.3. 输入变体

大多数 FT-TTS（微调）和 ZS-TTS（零样本）模型需要两个输入：（i）提供说话人、风格或声学参考的提示音频，以及（ii）待合成的目标文本。我们系统地改变这两种输入，以构建以下子集。

提示音频变体包含三个子集：(i)固定提示，在该子集中，整个子集重复使用同一段提示音频；(ii) 随机提示，每次合成都独立选择一段内容与目标文本不同的提示音频；(iii)同内容提示，提示音频与目标文本相匹配。除非另有说明，默认合成子集使用固定提示。

目标文本 变体包含两个子集：（i）13k 条真实 LJSpeech 话语的转录文本；以及（ii）由大语言模型生成的 5k 条转录文本，涵盖多个主题（科技、娱乐、地理、历史和政治）。默认合成子集和变体合成子集均包含 18k 条话语，这些话语取自上述真实文本和生成文本的并集。

3.2.4. 生成参数变体

我们通过系统地扫描各文本转语音家族特有的推理控制项，为每个文本转语音家族引入生成参数变体。具体而言，我们改变：(i)随机初始化（例如，基于扩散的模型所使用的随机噪声种子）；(ii) 推理步数，包括决定去噪轨迹的扩散步数或常微分方程求解器步数；(iii) 采样温度 $\tau \in [0.1, 1.0]$，用于调节输出随机性；(iv) 表现力/情感缩放 $\alpha \in [0, 1]$ 或离散情感标签，用于控制风格强度；以及 (v)时长/速度因子 r（其中 $r < 1$ 比默认值慢，$r > 1$ 比默认值快）。这些受控变化产生了一系列合成子集，可用于分离各参数对最终语音的影响。

虽然上述控制项具有普遍性，但各个文本转语音家族也提供了特定于模型的调节参数。例如，[19]不使用显式的情感标签或传统的温度参数；相反，它采用（α, β）来控制从提示音频中提取的情感/风格程度，并使用嵌入缩放参数来有效控制采样变异性。

3.2.5. 神经后处理变体

我们还使用五种先进的声码器和编解码器模型，对真实语音音频和 34 个默认合成子集进行神经后处理。我们考虑了两种变体：（i）重新声码化，即将每条话语转换为梅尔频谱图，再使用神经声码器进行重合成；以及（ii）编解码器重合成（重新编解码），即使用神经编解码器对从话语中提取的离散码进行解码，以重建波形。该变体生成了 10 个真实子集和 340 个合成子集。图 1 说明了清晰的处理流程：不同的生成式声学模型配置与声码器配对，以生成不同的声学-声码器变体，并对默认配置和架构进行神经后处理。

3.3. 元数据准备

鉴于该数据集的规模和复杂性，元数据准备至关重要。我们提供了两个互为补充的元数据文件，它们共享一个名为“UniqueID”的键列。第一个文件（文件索引）列出了每个样本的 UniqueID、绝对或相对文件路径，以及顶层标签（真实或伪造）。第二个文件（变体索引）将每个 UniqueID 与其生成细节关联起来，包括变体类别（例如，基于训练、基于架构、基于输入）、变体标签（特定变体的唯一标识符）、简短的变体描述（例如，温度 = 0.5）、

文本转语音系列	训练 监督式 微调 少样本			DS	输入架构 神经后置网络 总计 提示词 文本 声码器 步数 温度 情感 时长 重新声码 重新编码									
神经隐马尔可夫模型	√	1			✗	✗	4	5	✗	✗	✗	5	5	19
Tacotron2	✓	✗	✗	1	✗	✗	4	7	✗	✗	✗	5	5	21
FastSpeech2	√	1			✗	✗	4	5	✗	✗	✗	5	5	19
SpeedySpeech	✓	✗	✗	1	✗	✗	4	5	✗	✗	✗	5	5	19
FastPitch	√	1			✗	✗	4	7	✗	✗	✗	5	5	21
溢出	✓	✗	✗	1	✗	✗	4	5	✗	✗	✗	5	5	19
Mixer-TTS	√	1			✗	✗	4	4	✗	✗	✗	5	5	18
Glow-TTS	✓	✗	✗	1	✗	✗	4	5	✗	✗	✗	5	5	19
VITS	√	1			✗	✗	4	1	✗	✗	✗	5	5	15
STYLETTS2	✓	✓	✓	3	3	4	2	2	2	3	2	5	5	78
Matcha-TTS	√	1			✗	✗	4	4	2	3	✗	5	5	23
YourTTS	✗	✗	✓	1	3	4	✗	✗	3	✗	✗	5	5	20
Tortoise	√	1			3	4	✗	4	✗	✗	✗	5	5	21
VALL-E	✗	✗	✓	1	3	4	✗	✗	3	✗	✗	5	5	20
WhisperSpeech	√	1			3	4	✗	✗	3	✗	✗	5	5	20
XTTSv2	✗	✓	✓	2	3	4	✗	✗	3	✗	3	5	5	46
HierSpeech++	√	1			3	4	✗	✗	3	✗	✗	5	5	20
MaskGCT	✗	✗	✓	1	3	4	✗	✗	3	✗	3	5	5	23
FireRedTTS	√	1			3	4	✗	✗	3	✗	3	5	5	23
F5TTS	✗	✗	✓	1	3	4	✗	3	3	3	✗	5	5	29
E2TTS	√	1			3	4	✗	3	3	3	✗	5	5	29
FishSpeech	✗	✗	✓	1	3	4	✗	✗	3	3	✗	5	5	23
CosyVoice2	√	1			3	4	✗	✗	3	3	✗	5	5	23
Zonos	✗	✗	✓	1	3	4	✗	✗	3	3	3	5	5	26
Spark-TTS	√	1			3	4	✗	✗	3	3	3	5	5	26
MegaTTS3	✗	✗	✓	1	3	4	✗	3	3	✗	✗	5	5	23
OrpheusTTS	√	1			3	4	✗	✗	3	✗	3	5	5	23
MuyanTTS	✗	✗	✓	1	3	4	✗	✗	3	✗	3	5	5	23
OuteTTS	√	1			3	4	✗	✗	3	✗	3	5	5	23
ChatterboxTTS	✗	✗	✓	1	3	4	✗	✗	3	3	3	5	5	26
IndexTTS2	√	1			3	4	✗	✗	3	3	3	5	5	26

表2。文本转语音家族及每个家族的变体子集数量。各列概述训练机制（SS/FT/ZS）、默认子集数量（DS）、输入条件、架构/生成控制、神经后处理，以及每个家族最终的变体子集总数。图例：prom. = 提示音色；Text =目标文本；Voc. = 声学-声码器变体；Steps = 扩散步数；Temp = 采样温度；Emo =情感；Dur = 时长；Re-Voc. = 重新声码；Re-COD. =重新编解码。

TTS 模型名称、TTS 模型类型（例如，扩散、大语言模型、Transformer、扩散Transformer），以及一个用于指示样本属于默认子集还是变体子集的标志。

4. 分析范围与讨论

在完成数据生成后，我们界定了这一大型单说话人语料库所支持的分析范围，并强调其在受控、细粒度研究中的优势。我们的分析围绕三个相互正交的变化轴展开：训练类别（SS-/FT-/ZS-TTS）、输入来源（例如，真实语音脚本与生成文本、固定与随机参考音频），以及生成参数（例如，采样温度、扩散/求解器步数、时长/速度控制），此外还包括神经后处理（重新声码化和编解码器重合成）。

4.1. 为什么需要变体?

基于输入的变体，尤其是将生成文本与真实脚本配对，能够促使检测器在真实和经过操纵的语言内容之间实现泛化，而不是过拟合于仅包含“原始说出的话语”的样本。

训练类别变体有助于区分在单说话人上预训练的模型（SS-TTS）、针对目标说话人微调的模型（FT-TTS）以及零样本系统（ZS-TTS），这些系统使用几秒钟的参考片段，以此前未见过的说话人为条件。

生成参数变体使检测器接触同一合成器的多种实现方式；这些合成器在架构上相似，但配置不同，因此检测器不会过拟合于默认设置（例如，温度为 = 0.7）。这一点至关重要，因为攻击者可以调整此类参数，生成听起来更加自然的伪造语音，从而影响人们的观点。

尽管 [13, 21] 主张将经编解码器重合成的真实语音（即经过神经网络后处理、但其他方面未被更改的音频）视为伪造语音，以捕捉编解码器伪影，但我们采取不同的立场：我们将基于编解码器的文本转语音，以及重合成或重新声码器编码的合成语音标记为伪造语音，并将经编解码器重合成和重新声码器编码的真实语音保留为真实语音。反语音伪造问题的核心在于语音在上下文、声学、情感或音色方面遭到操纵；仅对真实语音信号进行重合成并不会改变这些属性中的任何一项。这一标注选择减少了对梅尔频谱图或声学编码解压缩所引入伪影的偏向，转而促使检测器关注在为经过操纵的内容构建这些表示时产生的失真和异常。

4.2. 划分方案

为训练当代反欺骗系统，我们对真实话语和伪造话语进行等量采样。我们提出了一种策略性划分方案，为每个 ZS-TTS 和 SS-TTS 分配 750 条话语，同时确保覆盖不同的变体家族。各行对应文本相似度条件：**TTT**（真实语音与生成伪造语音使用相同文本）、**TT** （TTS 默认子集之间使用相同文本）、**T** （变体子集之间使用相同文本）以及**t**（不同标签、文本转语音家族和变体之间的所有文本均不相同）。对于真实语音，划分方案平衡了原始语音与神经网络后处理语音，其中部分条目在不同标签之间共享文本，其他条目则使用未被重复用于生成伪造语音的内容。

5. 结论与未来展望

本研究推出了一个“首选”评测语料库，用于评估通用反欺骗系统抵御多样化合成攻击的稳健性；同时为说话人专属反语音伪造架构提供训练数据，并凭借丰富的伪造语音变体和每个子集充足的话语数量，为合成源追踪提供理想资源。该数据集将以开放源代码形式发布，以支持可复现研究并促进社区广泛采用。

我们将把语料库和分析扩展到真实场景语音和多位说话人，从而能够更深入、更全面地研究不同合成器及录音条件的影响。

6. 参考文献

[1] Kate Knibbs，“研究人员称，拜登深度伪造自动语音电话很可能使用了人工智能初创公司 ElevenLabs 的工具制作”，《连线》，2024年1月，栏目：标签。

[2] Xu Tan、Tao Qin、Frank Soong 和 Tie-Yan Liu，“神经语音合成综述”，2021年。

[3] Tianxin Xie、Yan Rong、Pengfei Zhang、Wenwu Wang 和 Li Liu，“大语言模型时代的可控语音合成：系统综述”，*arXiv* 电子预印本，第 arXiv:2412.06602 页，2024年12月。

[4] Anton Firc、Kamil Malinka 和 Petr Han´aˇcek，“Diffuse or confuse：扩散式深度伪造语音数据集”，2024年。

[5] Joel Frank 和 Lea Sch¨onherr，“Wavefake：用于促进音频深度伪造检测的数据集”，2021年。[6] Xin Wang、Hector Delgado、Hemlata Tak、Jee-weon Jung、Hye-jin Shim、Massimiliano Todisco、Ivan Kukanov、Xuechen Liu、Md Sahidullah、Tomi Kinnunen、Nicholas Evans、Kong Aik Lee 和 Junichi Yamagishi，“ASVspoof 5：大规模众包语音数据、深度伪造与对抗攻击”，2024年。[7] Junichi Yamagishi、Xin Wang、Massimiliano Todisco、Md Sahidullah、Jose Patino、Andreas Nautsch、Xuechen Liu、Kong Aik Lee、Tomi Kinnunen、Nicholas Evans 和 H´ector Delgado，“ASVspoof 2021：加速伪造语音与深度伪造语音检测的进展”，2021年。[8] Junichi Yamagishi、Massimiliano Todisco、Md Sahidullah、H´ector Delgado、Xin Wang、Nicholas Evans、Tomi Kinnunen、Kong Aik Lee、Ville Vestman 和 Andreas Nautsch，“ASVspoof 2019：第三届自动说话人验证欺骗攻击与对策挑战赛数据库”，爱丁堡大学 CSTR DataShare，https://doi.org/10.7488/ds/2555，2019年，访问日期：2025年6月30日。[9] Hashim Ali、Surya Subramani、Raksha Varahamurthy、Nithin Adupa、Lekha Bollinani 和 Hafiz Malik，“为知名人物收集、整理和标注高质量语音深度伪造数据集：流程与挑战”，载于*Interspeech2025*，2025年，第3928–3932页。

[10] Nicolas M. M¨uller、Piotr Kawa、Wei Herng Choong、Edresson Casanova、Eren G¨olge、Thorsten M¨uller、Piotr Syga、Philip Sperl 和 Konstantin B¨ottinger，《Mlaad：多语言音频反欺骗数据集》，2024年。

[11]Kratika Bhagtani、Amit Kumar Singh Yadav、Paolo Bestagini 和 Edward J. Delp，《Diffssd：一个基于扩散的语音取证数据集》，2024年。

[12] Jiawei Du、I-Ming Lin、I-Hsiang Chiu、Xuanjun Chen、Haibin Wu、Wenze Ren、Yu Tsao、Hung-yi Lee 和 Jyh-Shing Roger Jang，《Dfadd：基于扩散和流匹配的音频深度伪造数据集》，2024年。[13] Jiawei Du、Xuanjun Chen、Haibin Wu、Lin Zhang、I-Ming Lin、I-Hsiang Chiu、Wenze Ren、Yuan Tseng、Yu Tsao、Jyh-Shing Roger Jang 和 Hung-yi Lee，《Codecfake-Omni：大规模基于编解码器的深度伪造语音数据集》，2025年。[14] Anton Firc、Manasi Chibber、Jagabandhu Mishra、Vishwanath Pratap Singh、Tomi Kinnunen 和 Kamil Malinka，《STOPA：用于开放集来源追踪与归因的深度伪造音频系统性变化数据库》，2025年。[15]Keith Ito 和 Linda Johnson，《LJ 语音数据集》，https://keithito.com/LJ-Speech-Dataset/，2017年。[16] Shivam Mehta、Ruibo Tu、Jonas Beskow、Eva Sz´ekely 和 Gustav Eje Henter，《Matcha-TTS：一种采用条件流匹配的快速文本转语音架构》，载于*ICASSP 2024-2024 IEEE*声学、语音与信号处理国际会议（*ICASSP*）。IEEE，2024年，第11341–11345页。[17] Edresson Casanova、Kelly Davis、Eren G¨olge、G¨orkem G¨oknar、Iulian Gulea、Logan Hart、Aya Aljafari、Joshua Meyer、Reuben Morais、Samuel Olayemi 等，《Xtts：大规模多语言零样本文本转语音模型》，*arXiv*预印本 *arXiv:2406.04904*，2024年。[18] Yushen Chen、Zhikang Niu、Ziyang Ma、Keqi Deng、Chunhui Wang 和 等 Zhao，《F5-tts：通过流匹配生成流畅且忠实语音的童话讲述者》，*arXiv*预印本 *arXiv:2410.06885*，2024年。[19]Yinghao Aaron Li、Cong Han、Vinay S. Raghavan、Gavin Mischler 和 Nima Mesgarani，《Styletts 2：借助风格扩散、对抗训练和大型语音语言模型，迈向人类水平的文本转语音》，第36卷，2023年。[20]Dongchao Yang、Songxiang Liu、Rongjie Huang、Jinchuan Tian、Chao Weng 和 Yuexian Zou，《Hifi-codec：用于高保真音频编解码器的组残差矢量量化》，*arXiv*预印本 *arXiv:2305.02765*，2023年。[21] Yuankun Xie、Yi Lu、Ruibo Fu、Zhengqi Wen、Zhiyong Wang、Jianhua Tao、Xin Qi、Xiaopeng Wang、Yukun Liu、Haonan Cheng 等，《Codecfake 数据集及其用于通用深度伪造音频检测的反制措施》，*arXiv*预印本 *arXiv:2405.04880*，2024年。