WhisperAlign：面向孟加拉语长篇语音的词边界感知自动语音识别与以 WhisperX 为锚点的 Pyannote 说话人分离*

Aurchi Chowdhury计算机科学与工程，孟加拉国工程技术大学
aurchichowdhury07@gmail.com

Rubaiyat -E-Zaman计算机科学与工程，孟加拉国工程技术大学
rubaiyatzaman08@gmail.com

Sk. Ashrafuzzaman Nafees计算机科学与工程，孟加拉国工程技术大学
ashrafuzzamannafees08@gmail.com

arXiv:2603.04809v1 [cs.SD] 5 Mar 2026

[1]

摘要—**本文介绍了我们为 DL Sprint 4.0 提出的解决方案，旨在应对孟加拉语长篇语音识别（任务 1）和说话人分离（任务 2）这两项挑战。处理长篇、多说话人的孟加拉语音频，在语音活动检测、重叠语音和上下文保留方面面临显著困难。为解决长音频转录问题，我们采用 *whisper-timestamped* 实施了稳健的音频分块策略，使我们能够将精确且保留上下文的片段输入经微调的声学模型，从而实现高准确率转录。对于说话人分离任务，我们开发了一个集成处理流程，利用 *pyannote.audio* 和 *WhisperX*。我们方法的一项关键贡献，是基于竞赛数据集对 Pyannote 分割模型进行领域特定微调。这种适配使模型能够更好地捕捉孟加拉语会话动态的细微特征，并准确判定复杂的重叠说话人边界。我们的研究方法表明，在低资源环境中，将智能的带时间戳分块应用于自动语音识别，并对说话人分离进行有针对性的分割微调，可显著降低词错误率（WER）和说话人错误率（DER）。**

一、引言

处理孟加拉语等低资源语言中长达一小时的多说话人音频面临诸多挑战，因为标准模型难以保留上下文、处理重叠语音，而且缺乏可靠的语言工具。传统的定长音频切分经常会截断词语并引发幻觉，因此，我们的自动语音识别方法采用逐词时间戳，将音频划分为精确的语义片段，而无需依赖外部词典。此外，由于现有的说话人分离模型往往无法捕捉孟加拉语会话动态，我们直接针对该语言独特的声学特征对分割模型进行了适配。通过将这一适配模型与语音活动检测（VAD）相结合，我们的双流处理流程能够清晰地解决说话人重叠问题，并消除环境噪声，从而生成高度准确、互斥的音轨。

二、创新与贡献

A. 语义化词边界感知分块

标准的长篇自动语音识别方法通常依赖固定长度的音频切分，这往往会在话语中途截断单词，导致上下文丢失和模型产生幻觉。如正如既有研究所指出的，Bara´nski 等人（2025） [1] 表明，当音频片段超过模型 30 秒的解码窗口时，Whisper 的幻觉率会明显上升，因为系统会转而采用额外的启发式方法，从而产生虚假输出。

为解决这一问题，我们提出了一种计算开销低、独立完备的分块策略，无需外部对齐工具、发音词典或辅助模型。首先，Silero 语音活动检测会识别并提取语音区域，确保分块边界不会落在语音活动期间。随后，我们利用 whisper-timestamped，通过处理流程中已有的同一孟加拉语 Whisper 检查点的交叉注意力对齐头提取逐词时间戳，并采用序列匹配将这些时间戳映射到真实标注词上——保留真实标注中的正确拼写，同时借用 Whisper 的时间锚点；缺失的锚点则通过线性插值补齐。之后，音频会被严格按照词边界切分为多个片段，且每段均不超过 Whisper 的 28 秒安全解码时限。整个过程在单一处理流程中完成，除模型本身外不依赖任何其他工具，因此比蒙特利尔强制对齐器等强制对齐方法实用得多；这类方法需要特定语言的音素词典，而孟加拉语的此类词典并不容易获取。

B. 结合独占重叠处理与 *VAD* 交集的孟加拉语适配说话人分离

我们的说话人分离流程提出了三项精简贡献，以便在遵守严格竞赛规则的同时适配孟加拉语：

- **参数高效域适配：**我们并未重新训练整个庞大的说话人分离流程，而仅在孟加拉语音频上微调了 Pyannote 时间分段模型（∼1.4M 参数）。这有效地让模型学习了孟加拉语特有的对话韵律和轮流发言句法，同时将计算成本保持在较低水平。
- **算法互斥性：** 竞赛要求说话人轨道之间互斥。我们没有编写脆弱的 Python 脚本来手动删除或切分重叠片段（这会不可避免地破坏有价值的声学数据），而是使用了原生的独占式_说话人_分离功能。该功能

[1]GitHub 仓库

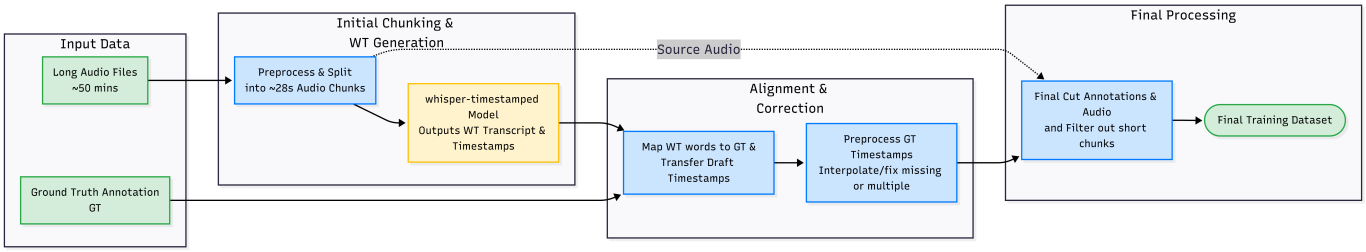


图 1. 端到端训练数据流水线：从原始长音频到对齐且尊重边界的分块，用于微调。

通过概率方式修改模型的输出，将重叠部分严格分配给最早出现的主导说话人，从而轻松最大限度地提高召回率。

•**WhisperX VAD 交集：**将 WhisperX 自动语音识别与独立的 Pyannote 说话人分离相结合，通常会导致时间漂移，因为它们依赖不同的语音活动检测（VAD）模型。 [2]我们通过执行逻辑与交集解决了这一问题，将 Pyannote 的聚类输出严格掩蔽到 WhisperX 所使用的 Silero 语音活动检测边界之内。这确保了文本与说话人标签之间的完美时间对齐，同时消除了环境幻觉。

孟加拉语长篇语音识别

III. 方法与架构

A. 系统概述

我们的系统由三个顺序阶段组成：训练数据准备、模型微调和推理。第一阶段通过构建一个自包含的分块并对齐处理流程，应对竞赛的核心数据挑战——所提供的长篇录音中缺少话语级分段——从而生成适合微调的干净（音频，文本）对。第二阶段在这些配对上微调一个领域特定的孟加拉语 Whisper 检查点。第三阶段使用带有幻觉后处理与双 GPU 并行的 VAD 引导的推理处理流程，将微调后的模型应用于未见过的测试音频。图 1 展示了整体流程。

B. 数据准备

竞赛提供了113段长篇孟加拉语录音，每段时长约一小时，并配有段落级文本标注。未提供话语级边界、说话人时间戳或音素对齐。因此，原始数据集与Whisper的输入要求存在根本性不匹配：Whisper仅处理最长30秒的片段，这使得稳健的分段策略成为处理流程中首要且最关键的一步。

分块。每段录音都经过多步骤分割处理流程。首先，应用 Silero 语音活动检测来检测并提取含语音区域，去除静音和背景噪声，并将剩余片段拼接成一段连续的语音波形。这样可以确保 Whisper 解码预算全部用于语音而非静音，从而降低在非语音区域产生幻觉的风险。

随后将清洗后的波形划分为28秒窗口，并由*whisper-timestamped*对每个窗口独立转写。该方法利用孟加拉语 Whisper 检查点的交叉注意力对齐头，无需任何外部对齐工具即可推导每个词的起止时间戳。由于时间戳仅在各窗口内局部有效，因此加入等于窗口起始时间的全局偏移量，将所有时间戳映射到完整音频时间轴，从而生成覆盖整段录音的扁平(word, global_start, global_end)元组列表。

随后，使用difflib.SequenceMatcher将这些带时间戳的词与真实标注对齐；该方法对Whisper的词序列与真实词序列进行差异比较。正确转写的词直接获得其时间戳；转写错误的词保留真实拼写，但借用Whisper对应词的时间戳；被Whisper完全遗漏的词，其时间戳通过相邻锚点之间的线性插值估计；而Whisper幻觉出、没有真实对应项的词则被丢弃。最终得到一条完整的带时间标注的真实词序列，其中每个词都在音频时间轴上带有可靠的位置锚点。

最后，该对齐词表被贪心地划分为连续片段：不断累积词语，直到加入下一个词会使片段超过28秒，此时便开启一个新片段。完成切分后，仅保留时长落在20至28秒范围内的片段用于微调。短于20秒的片段被丢弃，因为它们被发现无法提供足够的声学上下文——Whisper在极短输入上的性能会明显下降，此时模型可用于约束其解码的语音信号有限。随后对切分后的片段进行了一轮人工检查，以核验边界质量、确认未发生词中截断，并移除转录文本看似损坏或未对齐的配对。保留下来的片段被用作微调数据集，每个片段均保证以完整词语开始和结束，且其配对的文本标签与该片段内可听到的语音完全对应。

最终数据集统计。经过分块和筛选后，该

处理流程生成了约 4,553 个训练片段，平均时长为 26.56 秒，这些片段源自 113 份源录音。数据集使用固定的随机种子以确保可复现性，并按照 90/10 的比例划分为训练集和验证集。

表 I 分块自动语音识别数据集摘要

统计项	数值
分块总数	4,553
总时长	33.59 小时
平均片段时长	26.56 秒
最短片段	20.00 秒

C. 微调

我们在第一阶段生成的竞赛领域音频块上，对 bengaliAI/tugstugi_bengaliai- asr_whisper-medium 进行微调。该模型是 OpenAI 的 Whisper-medium（拥有 3.07 亿个参数）的孟加拉语专用微调版本。所有模型权重均进行端到端更新，未冻结任何层。解码器配置了强制语言和任务标记（语言=bn，任务=transcribe），并清空 suppress_tokens，以允许不受限制的孟加拉语词汇输出。

每个音频块都会转换为具有 80 个通道、填充至 3000 个时间帧的对数梅尔频谱图；文本标签则进行分词，并排除超过 444 个标记的样本，以确保不超出 Whisper 解码器的限制。训练采用教师强制，即解码器在每一步接收真实标记，而不是接收自身的预测结果——这比自回归生成高效得多，并能产生更稳定的梯度更新。标签填充位置填入-100，交叉熵损失按照惯例会忽略该值。

模型使用 AdamW 训练 5 个训练轮次，学习率为 1×10^{-5}，并设置 350 个预热步数。选择较为保守的学习率，是为了防止基础检查点中已经存在的孟加拉语表示发生灾难性遗忘。通过累积 2 个步骤的梯度，最终实现了 32 的有效批量大小。由于教师强制下的 logits 无法反映真实的生成质量，自定义的 ChunkWERCallback 会在每个训练轮次结束时，在验证集上执行完整的自回归生成，并在 Unicode 规范化后计算词错误率，同时保存表现最佳的检查点用于推理。

D. 推理

推理在 Kaggle 的双 T4 GPU 环境中进行。每段测试录音首先通过 Silero 语音活动检测以分离语音区域，这些区域被合并为最长 30 秒的片段，相邻片段之间有 1 秒重叠。每个片段由微调后的处理流程使用贪心解码进行转录，并设置 condition_on_previous_text=False，从而防止较早片段中的转录文本错误通过 Whisper 的条件机制传播到后续片段。

后处理对原始输出应用了两个过滤器：一个针对 Whisper 循环失效模式的 n 元语法重复检测器，以及一个用于移除源自 Whisper 的 YouTube 预训练数据的英语套话短语的字符串黑名单。这两个过滤器都在测试输出中捕获了真实案例。为了最大化吞吐量，通过 ThreadPoolExecutor 在两块可用 GPU 上加载了两个独立的模型实例，以轮询顺序分配文件，并实现了接近 2× 顺序单 GPU 处理的推理速度。

IV. 实验与评估

A. 评估指标

提交结果采用词错误率（WER）进行评估，该指标在文件级别计算预测转录文本与真实转录文本之间的差异。竞赛同时报告公开排行榜分数（在 29% 的测试数据上评估）和最终私有分数（在 71% 的测试数据上评估）。

B. 结果

表 II 展示了处理流程从原始基线到最终提交版本逐步改进后的词错误率变化。每一行代表一项单独的渐进式改动，用于单独考察各个组件的贡献。结

表 II 连续处理流程改进中的词错误率变化

系统	公开词错误率	私有词错误率
Tugstugi — 原始，未处理	0.675	0.702
+ 语音活动检测 + 后处理	0.419	0.440
+ Unicode 规范化	0.348	0.375
+ 微调（采用我们的分块策略）	0.265	0.296
+手动数据清理（最终版）	**0.252**	**0.278**

结果表明，每个阶段都带来了稳定且显著的改进。最大的一项提升来自采用我们的分块对齐策略进行微调，使公开词错误率降至 0.265——这证实了针对遵循边界的片段进行领域适配微调，是处理流程中影响最大的环节。通过人工检查分块边界，性能又进一步提升，这也证实了标注质量会直接影响模型性能。

V. 发现与分析

结果表明，处理流程的每个阶段都对最终性能作出了实质性贡献。单项增益最大的是在基础后处理的同时引入 VAD，使公开词错误率从 0.675 降至 0.419，且这一变化发生在进行任何微调之前。这表明，原始基线的错误中有相当一部分是由 Whisper 在静音和非语音区域上进行解码所触发的幻觉伪影——这也证实，对于长时段孟加拉语自动语音识别而言，干净的输入边界与模型质量同等重要。

将微调数据限制为 20–28 秒范围内的片段也产生了重要影响。较短的片段无法为 Whisper 编码器提供充足的声学上下文，从而导致梯度更新噪声更大、泛化能力更弱。仅保留较长且内容丰富的片段，确保模型能够在与测试时所面对的完整长度推理条件相匹配的样本上进行训练，这促成了在全部五个训练轮次中持续观察到的词错误率提升。

VI. 基准测试与比较

表 III 将我们的系统与文献 [3] 中报告的孟加拉语自动语音识别既有基线在 Bengali-Loop 测试集上的表现进行比较，同时列出我们在竞赛中的结果。

表 III 与既有孟加拉语自动语音识别基线的词错误率比较。

模型	词错误率 (%)
Hishab TITU-BN	50.67
Tugstugi（原始基线）	34.07
我们的方案（微调 + 完整处理流程）	27.00

尽管 Tugstugi 本身已是性能强劲的孟加拉语专用检查点，我们的系统仍比其原始基线的绝对分数高出约 7 分。这一提升归功于领域自适应：我们通过分块对齐流水线构建竞赛专用微调数据，并结合语音活动检测引导的推理和幻觉后处理。与具有竞争力的孟加拉语自动语音识别系统 Hishab TITU-BN 相比，我们的系统也取得了显著提升，绝对分数高出约 23 分——这凸显出，对于长音频转录而言，采用针对性微调的基于 Whisper 的架构优于传统的孟加拉语自动语音识别方法。

孟加拉语长音频说话人日志

VII. 方法与架构

该架构在逻辑上分为离线微调阶段和高度优化的推理序列。

A. 微调架构

- **冻结嵌入：**包含 660 万个参数的 wespeaker-voxceleb-resnet34-LM 嵌入模型完全保持冻结。
- **针对性分割训练：**微调仅针对轻量级 segmentation-3.0 模块（140 万个参数）进行。
- **数据与计算：**模型使用本地化孟加拉语 WAV 音频进行训练，并配有标注 CSV 文件，用于映射精确的开始_时间、结束_时间和说话人_ID 边界。在单台 Kaggle NVIDIA T4 GPU 上约 1 小时即完成收敛，最终生成了高度专用的孟加拉语检查点。

B. 主要推理笔记本策略

目标推理笔记本经过专门设计，兼顾速度与精度，使用 Kaggle P100 GPU 在 1.5 小时内处理完整个测试数据集。它按顺序执行以下步骤：

1）**智能音频预处理：**音频会重采样至 16 kHz，并进行峰值归一化，以获得稳定的动态范围。2）**Silero 语音活动检测预过滤：**WhisperX 的 Silero 语音活动检测（v4/v5）会去除非语音区域，以节省下游计算资源。为捕捉尾部孟加拉语音素，阈值采用非对称调节（起始=0.4，结束=0.25）。3）**核心说话人分离与聚类：**经过过滤的音频通过 speaker-diarization-community-1 流水线处理，并使用我们的微调分割检查点。凝聚聚类经过严格校准，以适应快速的多方对话（阈值=0.58，最短时长关闭=0.05s）。使用 exclusive_speaker_diarization 参数原生解决多标签重叠问题。

4）**自适应后处理：**

- **自适应合并：**不同于静态规则，片段合并间隔会根据声学密度，在 0.15 秒至 0.8 秒之间动态调整（以 0.4 秒为基准）。
- **瞬态过滤：**时长短于 0.15 秒的片段会被删除。
- **双重语音活动检测交集：**最终的 Pyannote 片段会与初始的 Silero 语音活动检测边界进行逻辑交叉核验。任何落在该逻辑与交集掩码之外的说话人分段片段都会被剔除，从而消除边界幻觉。

VIII. 实验与评估

最终的处理流程通过系统性的消融过程逐步构建。我们逐步测试架构调整，并根据竞赛的公开排行榜评估其表现。由于竞赛期间私有分数一直未公开，所有优化决策均严格依据公开排行榜的反馈作出。

A. 架构演进与指标进展

初始基线直接使用预训练的 speaker-diarization-3.1 模型，效果不佳，原因是其声学先验偏向西方语音。经过基础预处理后，基线有所改善。迁移到近期发布的 community-1 模型后，原生说话人分配效果大幅提升。不过，最显著的性能飞跃是在对分割模块进行微调，并最终加入 Whisper VAD 交集处理步骤以消除幻觉边界之后实现的。

我们的评估指标具体变化情况详见表 IV。（注：为确保完整性，私有分数为事后报告。）

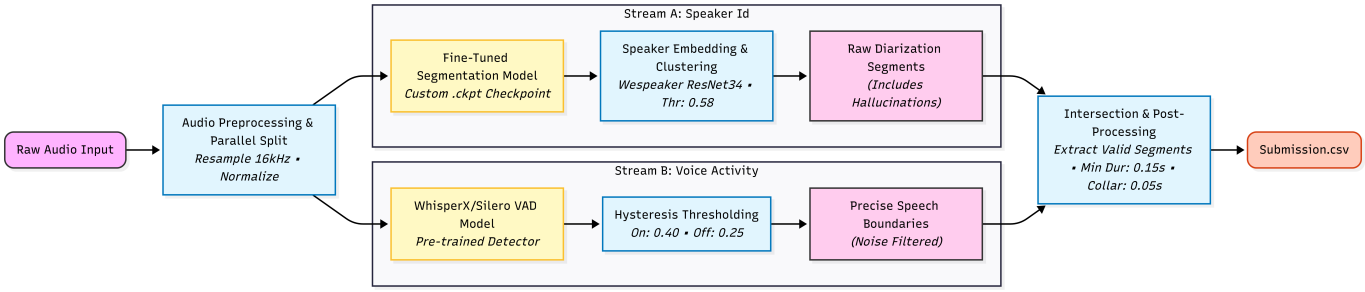


图 2. 所提出的并行说话人日志架构

表 IV 评估指标的进展

架构阶段	公开分数	私有分数
Pyannote 3.1 基础版	0.422	0.477
Pyannote 3.1 基础版 +预处理	0.364	0.413
处理		
Community-1 基础模型	0.314	0.376
Community-1 （微调	0.200	0.261
分割）		
Community-1（微调版） + VAD 交集	0.194	0.260

B. 通过网格搜索进行超参数优化

为最大限度提高最终准确率，我们针对处理流程的超参数进行了定向网格搜索，直接以公开排行榜上的指标为优化目标。

我们围绕 0.6 这一数值进行了局部搜索（测试了 0.6045、0.605 和 0.6055 等离散值）。最初，阈值为 0.605 时产生了最强的初步结果。然而，在对公开测试集进行进一步严格评估后，我们发现将阈值精确降至 0.58 能在公开排行榜上带来虽微小但明确的提升，因此将其作为我们最终选定的参数。

我们还发现，将容差参数严格设为 0.0 秒能取得更优结果。由于我们的处理流程主动利用排他式_说话人_分离，在原生帧级别以概率方式处理时间重叠，添加人为的事后容差会主动干扰神经网络精确的数学边界。强制采用零容差边界可确保最大的亚秒级精度，并与严格的竞赛约束保持一致。

IX. 发现与分析

我们的迭代开发过程产生了若干关键洞见：

分割模型的微调效率极高，仅用 20 个训练轮次便完全收敛（计算耗时约 1 小时）。最终的推理笔记本在 1 小时 25 分钟内处理完了整个包含 14 个文件的测试数据集，平均每个长音频文件仅需 6 至 7 分钟的处理时间。

由于语音转文本模型难以处理重叠语音，将语音转文本（Whisper）的时间戳与标准多标签说话人分离进行协调一直以来都极具挑战性。Pyannote 的专属_说话人_说话人分离通过确保任意时刻只有最可能被转录的说话人处于活动状态，从根本上解决了这一问题，大幅简化了对齐，并妥善解决了竞赛严格的非重叠规则。

独占说话人分离功能最初是为 Pyannote 的高级 precision-2 模型开发的。然而，precision-2 设有 30 小时的免费使用时长上限，这与竞赛的开源规则相冲突。近期发布的开放权重 community-1 模型使我们能够免费部署这一关键算法。

尽管微调和独占模式带来了巨大改进，但整合 WhisperX（Silero）VAD 交集才是促成我们最高总分的决定性因素。由于自动语音识别处理流程与 Pyannote 在内部从根本上依赖各自独立、彼此分离的语音活动检测模型，时间戳本质上并不匹配。将 Pyannote 的输出严格掩蔽在 Silero 精确的语音边界之内，彻底消除了这些边界幻觉。

升级到 community-1 引入了严格的依赖项要求，特别是需要更新版本的 NumPy。为防止执行失败，我们的推理笔记本经过专门设计，在每次推理运行中，于第一个依赖项安装单元格之后立即执行强制内核重启。

X. 基准测试与比较

评估复杂的 SD系统，需要针对语言专属数据集和全球数据集进行严格的基准测试。

*A. Bengali-Loop*基准（主要任务）

Bengali-Loop数据集 [3] 在本次竞赛所使用的数据集上建立了基准。它利用开箱即用的 pyannote.audio流水线，确立了 40.08% 的基础说话人错误率基线。在该基准测试中，传统模块化流水线严重劣化了音频，导致说话人错误率达到 61.50% 及以上。

表 V 在 Bengali-Loop 上评估的系统架构

系统架构	声学方法	来源	DER
WebRTC 语音活动检测 + ECAPA	模块化流水线	基线	73.71%
Silero 语音活动检测 + ECAPA	模块化流水线	基线	61.50%
Pyannote.audio（基础）	EEND 混合模型	基线	40.08%
所提出的双重语音活动检测	**微调 + 独占**	**所提出的方法**	**∼24% – 28%**

按照这些严格的学术标准，所提出的微调混合处理流程预计将使说话人错误率降低 12% 至 16%，相较于最先进的基础模型实现这一绝对错误率降幅。

B. 全球基准基线论证

为论证我们在微调前从 Pyannote 3.1 迁移至 community-1 的架构变更，我们评估了二者在标准全球数据集上的相对性能。与旧版 3.1 架构相比，community-1 模型在多样化的声学环境（例如正式会议、复杂的多样化音频以及电话语音）中始终表现出更低的错误率，证明了其作为基础模型的优越性。

表 VI 全球基准上的基础模型比较

基准数据集	领域	旧版（3.1）说话人错误率	Comm-1 说话人错误率
AMI（IHM）	会议	18.8%	17.0%
DIHARD 3（完整）	多样化	21.4%	20.2%
CALLHOME（p2）	电话	28.5%	26.7%
VoxConverse (v0.3)	网络视频	11.2%	11.2%

XI. 结论

本文提出了一种高度优化的双阶段流程，用于低资源长篇孟加拉语语音处理，成功解决了语义自动语音识别分块与复杂说话人分离两大问题。

针对长篇自动语音识别，我们提出了一种自包含的分块与对齐处理流程，利用逐词时间戳转移和序列匹配，从长录音中自动导出话语级训练片段，从而无需外部对齐工具或人工标注。在这些片段上微调 tugstugi Bengali Whisper检查点，并结合 VAD 引导的推理与幻觉后处理，使公开排行榜上的词错误率从 67.5% 降至 25.2%。

对于说话人分离，通过对分割模块进行参数高效微调，系统得以学习孟加拉语韵律；同时迁移至community-1架构，并原生利用独占_说话人_说话人分离，在无需破坏性人工启发式规则的情况下，解决了严格的非重叠约束。最后，通过将 Pyannote 的聚类结果与 WhisperX 的 Silero 语音活动检测进行逻辑交集运算，并强制采用零秒领圈，我们消除了时间边界漂移和环境幻觉。最终，这一计算高效的框架大幅超越了基线 Bengali-Loop基准，为在低资源环境中扩展语音技术提供了稳健且高度准确的蓝图。

参考文献

[1] M. Bara´nski、J. Jasi´nski、J. Bartolewska、S. Kacprzak、M. Witkowski 与 K. Kowalczyk，“非语音音频诱发的 Whisper 自动语音识别幻觉研究”，载于 *ICASSP 2025 - 2025 IEEE*声学、语音与信号处理国际会议 *(ICASSP)*，2025，第 1–5 页。[2] M. Bain、J. Huh、T. Han 与 A. Zisserman，“WhisperX：长音频的时间精确语音转录”，*INTERSPEECH 2023*，2023。[3] H. M. S. Tabib、I. A. Rifti、A. M. A. Ehsan、S. Dasgupta、M. Z. M. S. Sowdha、A. J. Sarker、M. R. I. Nijamy、T. Hossain、M. M. Khatun、M. Mahmood、R. Debnath、G. Biswas、A. Karim、W. A. A. Navid、M. Muztahid、F. A. Udoy、S. S. Rahman、M. T. R. Shifat、M. S. Khatun、M. Rahman、M. M. Hasan、A. Saha、M. N. M. Nobo、S. Bhattacharjee、T. Bhomik、A. N. Swapnil 与 S. Kabir，“Bengali-Loop：面向长篇孟加拉语自动语音识别与说话人分离的社区基准”，2026。[在线]。可获取：https://arxiv.org/abs/2602.14291