基于端侧语音识别的隐私保护智能笔记系统设计与实现

作者：韦治国（独立开发者） 邮箱：weizhiguo666888@163.com

摘要

智能笔记应用如今广泛采用云端识别方案：用户的麦克风音频被持续上传至第三方服务器，识别完成后再返回文字。语音同时是生物特征标识：声纹可以唯一锁定一个人。《中华人民共和国个人信息保护法》[13]将生物识别信息列为敏感个人信息，要求"具有特定的目的和充分的必要性，并采取严格保护措施"方可处理；《中华人民共和国数据安全法》[14]亦确立了数据分类分级保护的基本制度。在这一约束下，"音频不出设备"的端侧识别是一条值得验证的技术路线。

本文设计并实现了隐私保护智能笔记系统 AI Note。系统的核心设计是端云边界架构：语音活动检测、语音识别、声纹嵌入提取、语音降噪等全部音频处理均在手机本地完成，音频数据永不出设备；仅当用户主动调用摘要、问答、思维导图等文本智能功能时，文本内容才经自建中转服务转发至大语言模型，中转服务提供邮箱验证码认证、月度配额、滑动窗口限流与模型白名单等安全机制。

针对离线整段识别模型无法直接提供实时体验的问题，本文提出一种基于 VAD 状态机与预滚缓冲的流式化切窗策略，在不损失识别精度的前提下实现准实时转写；针对传统说话人分离须等待录音结束的问题，设计了逐段声纹嵌入的在线说话人聚类方法，使说话人标注在录音过程中实时产生。实验表明：①切窗策略在整个参数网格上保持了整段识别的精度（干净语音 CER 介于 2.98%~4.17%），0.6 秒预滚配合 1.0 秒停顿阈值取得最低字错率 2.98% 且句首缺字率为零；②在线说话人聚类在 31 段多人对话（含 20 段 AliMeeting 公开真实 3~4 人会议）上取得 76.6% 的归属准确率，比固定阈值基线高 7.8 个百分点，会后毫秒级离线精修进一步提升至 79.7%，且录音停止时无需任何额外等待；③int8 量化模型与 fp32 模型精度相当（平均 CER 3.23% 对 3.28%），实时率更优（0.344 对 0.380），模型体积仅为后者的四分之一（230MB 对 936MB）；④面向文本级隐私缺口，用"真实会议蒸馏、合成扩充、负例边界"三层数据对 1.7B 小模型做领域适配，经"缺陷发现、定向扩数据、复训验证"一轮迭代，编造待办与编造负责人的不犯率从 20% 提升至 80%，数字 token 级保真率达 98%，量化后模型 1.03GB，可在纯 CPU 上运行。真机 66 分钟全负载连续录音耐力测试无崩溃、无丢音，内存缓增平稳，电量消耗低于电量计 1% 分辨率。

本系统已在真实 Android 设备上完整实现，并通过 66 分钟全负载真机耐力测试，表明在当前主流手机上，音频处理全端侧化的隐私保护笔记方案是可行的。

关键词：端侧语音识别；隐私保护；说话人分离；智能笔记；大语言模型；参数高效微调

Abstract

With the spread of large language models and speech recognition, intelligent note-taking applications widely adopt cloud-based recognition: the user's microphone audio is continuously uploaded to third-party servers. Speech is not only a content carrier but also a biometric identifier. This paper presents AI Note, a privacy-preserving intelligent note-taking system built around an explicit edge-cloud boundary: voice activity detection, speech recognition, speaker embedding extraction and noise suppression all run entirely on the mobile device, and audio never leaves the phone; only when the user explicitly invokes text-level intelligence (summarization, question answering, mind-map

generation) does the text transit a self-hosted relay service that enforces authentication, monthly quotas, sliding-window rate limiting and model whitelisting. To deliver a real-time transcription experience with an offline whole-utterance ASR model, we propose a streaming-simulation segmentation strategy coupling a VAD state machine with a pre-roll ring buffer and dual-threshold window cutting; to annotate speakers without waiting for the recording to finish, we design an online speaker clustering method based on segment-level embeddings. Experiments show that: (1) the segmentation strategy preserves whole-utterance accuracy across the entire parameter grid (CER 2.98%–4.17% on clean speech), with a 0.6 s pre-roll and 1.0 s pause threshold achieving the lowest CER of 2.98% and zero onset truncation; (2) online speaker clustering attains 76.6% attribution accuracy on 31 multi-speaker conversations — including 20 real three-to-four-speaker meetings from the public AliMeeting corpus — 7.8 percentage points above the fixed-threshold baseline, and a millisecond-scale post-meeting refinement further reaches 79.7%, with zero added latency when recording stops; (3) the int8-quantized model matches the fp32 model in accuracy (3.23% vs. 3.28% mean CER) at a better real-time factor (0.344 vs. 0.380) and one quarter of the model size (230 MB vs. 936 MB); (4) a domain-adapted 1.7B meeting-minutes model, trained on a three-layer recipe of real-meeting distillation, synthetic expansion and boundary negatives and improved through one defect-driven data iteration, raises its clean rate on fabricated to-dos and fabricated owners from 20% to 80% at 98% token-level digit fidelity, and quantizes to 1.03 GB. A 66-minute full-load on-device endurance test completed without crash or audio loss, with memory drifting slowly and battery drain of only 3% over the full run. The complete system demonstrates that fully on-device, privacy-preserving intelligent note-taking is practical on commodity smartphones.

Key words: on-device speech recognition; privacy preservation; speaker diarization; intelligent note-taking; large language model

---

第一章 绪论

1.1 研究背景与意义

课堂、会议、采访，是学生、记者、律师等人群每天都要面对的记录场景。手写太慢，录音回去再听又极其低效：一小时录音就要花一小时回听。语音转文字笔记因此成为刚需，讯飞听见、腾讯云语音识别等云端方案已经相当成熟。

但成熟方案有一个很少被讨论的问题：**语音被上传了。**语音不只是"内容的载体"，它同时是生物特征——声纹像指纹一样可以唯一锁定一个人。音频一旦离开设备，用户对它的控制就结束了：服务器是否留存、留存多久、会不会被二次利用，用户都无从知晓。从合规角度看，《个人信息保护法》第二十八条明确将生物识别信息列入敏感个人信息，处理敏感个人信息必须具有特定目的和充分必要性并采取严格保护措施；《数据安全法》确立了数据分类分级保护制度，"最小必要"已经成为数据处理的明确导向。云端识别"为了转一段文字而上传全部音频"的做法，难以满足这一最小必要要求。

与此同时，端侧推理的条件已经成熟：①量化技术把几百兆的识别模型压到两百兆上下，主流手机芯片可以实时运行，端侧超小模型甚至能压进几十兆（45M 参数的 needle 量化后仅 14MB 单文件 [16]）；②sherpa-onnx 等嵌入式推理框架把语音识别、语音活动检测、声纹提取等能力打包到了 Android 平台；③轻量降噪模型（如

GTCRN）只有零点几兆。谷歌官方参考应用 AI Edge Gallery [17] 与 macOS 商用听写产品 FluidVoice [18] 的出现进一步表明，"全部音频处理都在手机本地完成"已经从实验室走向产品。

本文的意义因此有两层：①实践层面，为记者、律师、医生、学生等隐私敏感人群提供一个"音频不出手机"的可用工具；②方法层面，回答一个系统工程问题：当识别、分离、降噪全部搬到端侧之后，离线模型的实时性、说话人标注的等待、移动端算力的限制这三个拦路虎怎么解决，本文给出经过实验验证的方案。

1.2 国内外研究现状

端到端语音识别。 本节的模型均建立在 Transformer 架构[11]之上。Whisper[1] 用 68 万小时弱标注数据证明了大规模训练的多语言鲁棒性；Paraformer 的非自回归设计在中文工业界广泛部署。SenseVoice-Small 是阿里 FunAudioLLM 发布的多语言（中/英/日/韩/粤）编码器模型，内置标点预测与逆文本规范化，并官方提供 int8 量化导出，天然适合移动端。这类模型的共同特点是"整段识别"：必须拿到一段完整音频才输出结果，没有流式假设，这正是本文要解决的第一个矛盾。

语音活动检测。 Silero VAD 是一个基于 LSTM 的紧凑检测器，在 16kHz 音频上以 512 采样点为窗工作，提供起判/止判迟滞参数，常被用作识别前的前置分割器。本文把它作为"门控"使用，在其上叠加自己的缓冲与切窗逻辑。

说话人分离。 主流范式是 pyannote 为代表的离线管线：分割模型切分、逐段提取声纹嵌入、全量层次聚类，整个流程必须等录音结束后对完整音频执行。CAM++ 等嵌入模型提供紧凑的句级向量，余弦相似度可分性强。在线分离（如 UIS-RNN 及在线聚类变体）在学术界已有研究，但极少落地到移动端，手机单线程执行器跑不动分割模型。本文的在线逐段聚类是在移动端约束下对在线聚类的务实改造。

大模型文本智能与隐私。 大语言模型已广泛用于转写文本的摘要、问答与结构化。主流产品把全文直接发给服务商。本文的中转架构在用户与上游之间插入一个认证、配额、限流的中介层，使用户无需持有任何第三方密钥，也让文本上云成为显式、可审计的用户动作。另一条路线是把文本智能也搬回端侧：LoRA 等参数高效微调方法 [19] 让 1~2B 级开源模型（如 Qwen3 [20]）在单卡上分钟即可完成领域适配，GGUF 量化工具链能把适配后的模型压到 1GB 量级在纯 CPU 上运行。第六章实验六沿这条路线做了完整验证。

端侧智能系统路线。 谷歌 AI Edge Gallery [17] 以 LiteRT 在 CPU/GPU/NPU 多后端运行 Gemma int4，并用 270M 的 FunctionGemma 证明小模型可以承担端侧 function-calling，是"全本地"路线的代表；macOS 听写产品 FluidVoice [18] 以端侧 120M 识别模型加端侧增强小模型提供媲美云端的听写体验，其"识别免费 + 端侧增强收费"的分层也验证了端侧小模型的商用可行性。本文的混合架构与全本地路线形成对照：音频处理同样全部本地，但文本智能走云端中转，以有界、用户显式发起的文本披露换取更强的文本智能质量与零端侧大模型负担。在端云路由方向上，needle [16] 展示的置信度门控机制（低置信度请求才转云端）为本文的隐私分级路由提供了直接的可行性佐证。

1.3 本文主要工作

①设计并实现了端云边界的隐私保护架构：VAD、识别、声纹、降噪全部端侧完成，仅用户主动提交的文本经自建中转服务上云，中转服务具备邮箱验证码认证、月度配额、滑动窗口限流与模型白名单。

②提出离线整段识别模型的流式化切窗策略：Silero VAD 状态机门控（0.1 秒起判确认）、0.6 秒预滚环形缓冲补偿 VAD 起判滞后、双阈值切窗（0.6 秒停顿且段长不小于 1.5 秒，或达到 10~15 秒段长上限）、1.5 秒间隔快照解码提供识别中预览。

③设计逐段声纹嵌入的在线说话人聚类方法：每个切窗段解码后同步提取 CAM++ 嵌入，与已有说话人质心做余弦相似度匹配（阈值 0.6、上限 8 人、质心滑动平均更新），并以时长感知规则抑制短段过聚类、有界延迟重判（3 段前瞻动态规划平滑 + 切换惩罚）修正边界误判；说话人标签随录音实时产生（固定约 3 段延迟），停止录音零等待，会后还可用缓存嵌入做毫秒级离线精修。

④完整实现 Android 应用与 FastAPI 中转服务，并通过系列对照实验与真机测试验证：切窗参数网格实验、说话人聚类消融与超参数敏感性实验、嵌入模型与大型识别模型参照实验、模型配置权衡实验，以及 12 分钟全负载真机稳定性与耗电测试。

⑤端侧会议纪要小模型的领域适配与迭代：提出"真实会议蒸馏、合成扩充、负例边界"的三层数据配方，用缺陷驱动的定向扩数据完成一轮"训练、审读、扩数、复训"闭环，编造待办与编造负责人的不犯率从 20% 提升到 80%，模型量化至 1.03GB 并在纯 CPU 上跑通（第六章实验六）。

1.4 论文组织结构

第二章介绍相关技术；第三章给出系统分析与总体设计，重点是端云边界；第四章阐述两项关键方法与部署优化；第五章描述系统实现中的工程细节；第六章报告实验与真机测试；第七章总结并展望。

第二章 相关技术

2.1 sherpa-onnx 端侧推理框架

sherpa-onnx 是新一代 Kaldi 团队维护的开源推理框架，基于 ONNX Runtime，把语音识别、语音活动检测、说话人嵌入、语音增强等能力封装为 Android/iOS 可直接调用的 API。本系统 App 端使用 1.12.1 版本，实验脚本使用 1.13.4 版本，二者底层同为 ONNX Runtime，模型文件完全通用。

2.2 SenseVoice 多语言识别模型

SenseVoice-Small 支持中、英、日、韩、粤五种语言，内置标点预测与逆文本规范化（识别结果直接带标点、数字自动转为书写形式），官方同时发布 fp32（约 936MB）与 int8 量化（约 230MB）两个 ONNX 导出。它是编码器结构的整段识别模型：输入一段完整音频，输出整段文字，没有任何中间假设输出。精度好、体积小，但不提供流式体验，这是第四章切窗策略要解决的问题。

2.3 Silero VAD 语音活动检测

Silero VAD 是一个 LSTM 小模型，每次只接受恰好 512 个采样点（16kHz 下 32 毫秒），内部维护状态，输出当前窗是否语音。它暴露起判确认时长（minSpeechDuration）等迟滞参数：连续语音超过阈值才判定为"说话开始"，避免瞬时噪声误触发。本文用 v5 版本模型，内置在 App assets 中（版本选择的工程原因见 5.1 节）。

2.4 CAM++ 声纹嵌入模型

CAM++ 是面向说话人确认的轻量网络，把一段任意时长语音压缩为一个 192 维向量。向量的几何性质很好：同一说话人不同句子的向量方向接近，不同说话人方向远离，因此用余弦相似度即可度量"两段声音像不像同一个人"。模型仅 28MB，端侧逐段提取代价是一次前向计算。

2.5 GTCRN 轻量语音降噪

GTCRN 是 2024 年提出的超轻量语音增强模型，模型文件仅约 0.5MB，在保持降噪能力的同时计算量远低于更大的 CRN 变体，使"每个切窗段识别前先过一遍降噪"在手机上成为可能。本系统把它作为识别链路的可选前处理：只作用于送识别的音频副本，不影响录音原声。

2.6 OpenAI 兼容协议与 SSE 流式输出

中转服务对 App 暴露 OpenAI 兼容的 /v1/chat/completions 接口，上游默认接 DeepSeek[12]。回答通过 SSE（Server-Sent Events）流式下发：HTTP 长连接上逐段推送生成的文字，App 端打字机式显示，避免用户面对长时间白屏。

2.7 Android 技术栈

App 使用 Kotlin + Jetpack Compose（Material 3）构建界面，Hilt 依赖注入，Room 做本地持久化，DataStore 存配置，Retrofit/OkHttp 负责网络。录音基于 AudioRecord 采集 16kHz 单声道 PCM。

第三章 系统分析与总体设计

3.1 需求分析

功能性需求：①语音转文字笔记；②会议模式（实时字幕 + 说话人区分）；③AI 文本功能（摘要、问答、思维导图、周报、待办提取）；④数据管理（本地存储、导出、WebDAV 备份、账号与配额）。

非功能性需求：①隐私：音频与转写原文不出设备；②实时性：转写字幕准实时显示，说话人标注随录音产生；③离线可用：无网环境下录音转写完整可用，仅 AI 功能降级；④资源约束：单线程执行器下识别、嵌入不互相抢资源，长时间录音不崩溃。

3.2 端云边界设计（本章重点）

端云边界是本系统最核心的设计：音频不越界，文本仅在用户指令下越界。

数据分两级：①设备级（device-only）：麦克风音频、转写原文，代码层面就不存在上传音频的网络接口，WebDAV 备份也是用户自行配置的私有云；②中转瞬态级（relay-transient）：用户主动提交的提问文本与检索出的笔记片段，经中转服务转发上游，服务侧不落地存储，只保存配额计数器等元数据。这套分级直接映射《个人信息保护法》的"最小必要"原则：要提供摘要功能，必须处理文本，但没有任何理由处理音频，那就让音频一步都不出设备。

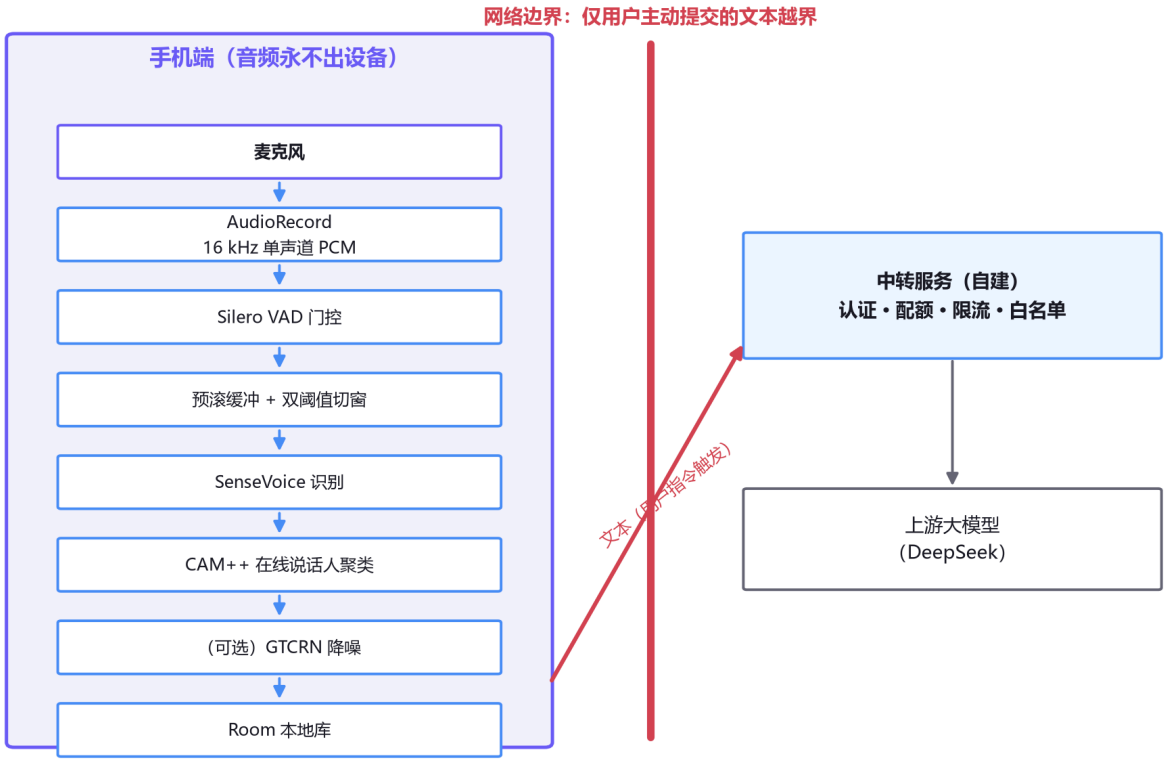

图 1：系统总体架构图。 左侧为手机端处理管线：麦克风 → AudioRecord（16kHz 单声道 PCM）→ Silero VAD 门控 → 预滚缓冲 + 双阈值切窗 → SenseVoice 识别 → CAM++ 在线说话人聚类 →（可选）GTCRN 降噪 → Room 本地库，音频全程不出设备；中间粗实线为网络边界，仅用户主动提交的文本越界；右侧为自建中转服务（认证/配额/限流/白名单）与上游大模型（DeepSeek）。

这一架构同时解决了三个实际问题：①隐私：生物特征不离端；②可用性：无网时核心功能照常；③成本结构：端侧识别没有按分钟计费，长会议随便录。

3.3 系统模块划分

App 端六个模块：采集模块（AudioRecord、前台服务保活）、识别模块（VAD 门控 + 切窗 + 解码）、说话人模块（逐段嵌入 + 在线聚类）、AI 模块（中转客户端、SSE 流式）、存储模块（Room 数据库）、同步与账号模块（WebDAV、邮箱码登录）。服务端一个模块：FastAPI 中转服务（认证、配额、限流、白名单、SSE 转发）。

3.4 数据库设计

本地 Room 库主要表：笔记表（标题、正文、类型、时间戳）、段落表（笔记外键、起止时间、说话人标签、文本）、AI 结果表（类型、内容、关联笔记）、待办表、问答历史表。服务端 SQLite 三张表：用户表（邮箱、配额余额、配额重置月）、验证码表（哈希、过期时间、尝试次数）、请求日志元数据表。服务端不存任何用户内容文本。

第四章 关键技术研究与实现

4.1 离线整段识别模型的流式化切窗策略

问题。 SenseVoice 只能整段解码：拿一段完整音频，出一段文字。直接把它用于边录边显会撞上三堵墙：①没有中间结果，用户录完之前屏幕是空的；② naive 定长切窗会在窗边界切断音素，精度受损；③把静音段送进去，模型会"幻听"——凭空编造不存在的文字。

方案。 采集线程以 100 毫秒（1600 采样）为一帧处理音频流，维护三样东西：一个装着最近 10 帧（0.6 秒）的预滚环形缓冲、一个当前活动段缓冲、连续非语音帧计数器。逐帧逻辑如下：

①帧先缓冲成 512 采样窗送 Silero VAD，连续 0.1 秒语音才确认起判（状态机从静音态转语音态）；

②非语音且没有活动段时，帧只留在环形缓冲里，不送识别——这是防"幻听"的第一道闸；

③起判瞬间，把环形缓冲里的 0.6 秒垫在段首。VAD 确认说话本身有滞后（实测约 0.25 秒以内），没有预滚，每段话的开头几个字就会被吃掉；有了预滚，起判滞后永远有缓冲兜底；

④切窗满足两个条件之一即触发：连续 6 帧（0.6 秒）非语音且段长不小于 1.5 秒（自然停顿切）；段长达到上限（默认 10 秒，长上下文模式 15 秒，强制切）；

⑤段内语音不足 0.3 秒的整段丢弃，视为咳嗽、碰麦克风之类的噪声；

⑥活动段每 1.5 秒做一次快照解码，作为"识别中"预览刷新到屏幕上（解码器忙则跳过）。

解码在单线程执行器上串行执行：段解码、快照解码、嵌入提取排队进行，避免移动芯片上的资源争抢。这套策略的效果是：用户看到的是 1.5 秒刷一次的滚动字幕（感知实时），段落终稿在停顿后约 3.5 秒落定（整段模型的物理下限），而精度与整段识别基本持平，第六章实验一给出了网格证据。

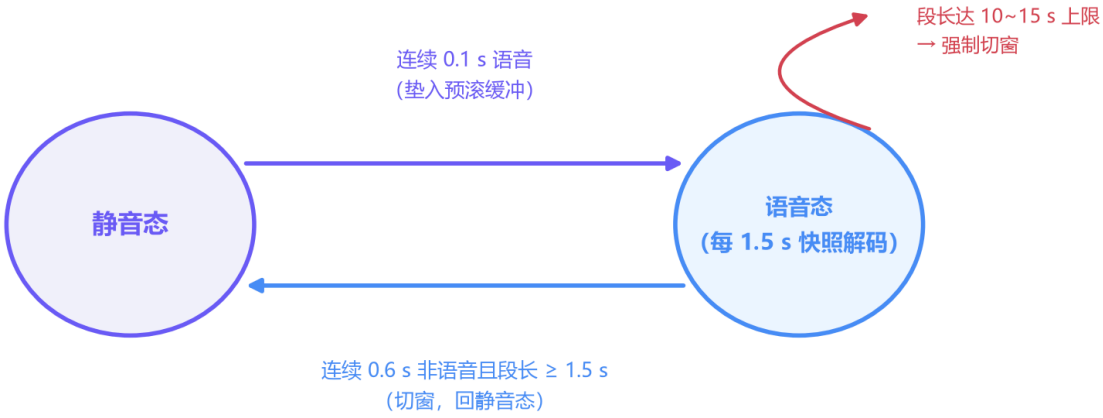

图 2：切窗状态机转移图。 静音态与语音态两个状态：连续 0.1s 语音进入语音态（垫入预滚缓冲）；连续 0.6s 非语音且段长 ≥1.5s 时切窗回静音态；段长达 10~15s 上限强制切窗；语音态内每 1.5s 提交一次快照解码。

4.2 逐段声纹嵌入的在线说话人聚类

问题。 pyannote 式离线分离要先用分割模型过一遍完整录音，再提嵌入、全量聚类，分钟级会议录完还要再等数十秒，而且分割模型根本跑不进手机单线程执行器。

方案。 每个切窗段解码定稿后，同步用 CAM++ 提取 192 维嵌入 e。系统维护说话人质心列表 $c_1 \ldots c_k$ 及各自命中次数 $n_1 \ldots n_k$，逐段增量分配：

```
j* = argmax_j cosine(e, c_j)
若 cosine(e, c_{j*}) ≥ τ(0.6)   → 该段归入说话人 j*
                                  质心滑动平均更新，n_{j*} 加一
否则若 k < K_max(8)             → 新建说话人，质心即 e
否则                            → 归入最近质心，不更新
```

这个设计有三个务实之处：①不依赖分割模型：VAD 切窗的段边界直接充当话轮假设，省掉了手机跑不动的部件；②每段只多一次前向计算，代价亚秒级，而且摊销在录音过程中，停止录音那一刻标注已经完成，零等待；③嵌入模型缺失时自动降级为无标注转写，功能不崩。准确率与延迟的实测对比见第六章实验二。

在上述朴素规则之上，最终算法还有四个组件，均在第六章实验二中逐项消融验证：

①**时长感知分配。**不足 1 秒的段直接继承上一说话人，不开簇也不更新质心；1~2 秒的段可以归入已有簇但不允许开新簇；只有 2 秒及以上的段参与建簇与质心更新。瞄准的是朴素规则的主要败因：短段和跨话轮边界段不断开出伪簇。

②**有界延迟重判。**缓存最近 N=3 个段，对窗口内的联合标注做动态规划平滑（代价 = 负相似度之和 + δ(0.05)×说话人切换次数），只提交窗口中最老的段。标注延迟因此固定为 3 段，不随录音时长增长。

③**自适应阈值。**接受阈值跟随已接受相似度的运行分布：clamp(μ−1.5σ, 0.45, 0.75)，前 6 个样本冷启动用固定 0.6。第六章的实验会表明这一组件单独使用没有收益，并给出幸存者偏差的分析。

④**会后离线精修。**录音停止后，用会议期间缓存的嵌入做二次聚类：以在线划分为初始结果，层次聚类（average linkage、余弦距离）重聚，轮廓系数在 [在线簇数−2, 在线簇数+1] 区间内选 k，再做 2 轮质心精炼，并合并质心余弦相似度超过 0.6（与在线阈值同源）的相似簇。嵌入全部复用，整个步骤只有毫秒级开销。

4.3 端侧模型兼容与部署优化

双模型可切换。 int8（230MB）为默认，fp32（936MB）供追求极致精度的用户下载。第六章实验三证明 int8 精度与 fp32 相当，这个默认是有数据支撑的。

模型下载与校验。 模型首用时从国内可达镜像（ModelScope、hf-mirror）多源兜底下载，带体积校验。嵌入式 ONNX Runtime 只支持有界的 ONNX IR 版本，模型加载时解析 protobuf 头的 ir_version 字段做校验，不合格的模型拒绝加载并让对应功能降级，而不是崩溃。

线程模型。 单线程执行器串行化解码与嵌入，牺牲一点并行度换取移动芯片上的稳定时序，避免识别和嵌入互相抢核导致的字幕卡顿。

4.4 中转服务的安全设计

中转服务（FastAPI + SQLite）部署在入门级云主机上，安全机制四道闸：

①**认证**：无密码设计。六位邮箱验证码（10 分钟有效、5 次尝试、每地址每天 10 条）换 30 天 JWT，App 后续请求带 JWT 即可；

②**配额**：每用户每月默认 100 次，请求时原子扣减，上游调用失败自动退还，用户不为服务故障买单；

③**限流**：滑动窗口每用户每分钟 20 次，单请求正文上限 6 万字符，防止滥用刷爆上游额度；

④**白名单**：模型精确匹配白名单，防止有人拿中转服务调昂贵模型套利。

上游 API 密钥、JWT 密钥、SMTP 凭据全部只存在于服务端环境文件，App 与代码仓库中无任何秘密。文本在转发瞬间技术上经过服务器内存但不落盘，这是"中转瞬态"分级的含义，也是第七章把端侧大模型摘要列为未来工作的原因。

第五章 系统实现

5.1 Android 端

App 用 Kotlin + Jetpack Compose（Material 3）实现，功能包括快速笔记、会议模式（实时字幕 + 说话人标注）、录音中实时 AI 纪要、回放（进度拖动、变速）、待办提取与任务中心、Markdown/TXT 导出、PIN 应用锁、SSE 流式问答与历史、WebView 离线渲染 mermaid 思维导图、AI 周报、WebDAV 备份、邮箱码登录与配额展示。两个工程细节值得单独写，因为它们都是踩坑换来的。

VAD 模型版本闪退排查。 开发期遇到一个典型的端侧部署陷阱：Silero VAD 模型导致 App 原生层崩溃（native SIGABRT），Java 层 try-catch 根本接不住。排查发现是 ONNX IR 版本兼容问题：onnx-community 发布的原版模型是 IR v10，超出 App 内嵌 ONNX Runtime 的支持范围，加载即崩；换 sherpa 官方 release 的 v4 模型，又报 "Unsupported silero vad model" 同样闪退。最终方案：把 v5 模型的 IR 版本降为 9（不改动任何权重）内置到 App assets，彻底去掉网络依赖和这条崩溃路径；同时在 isVadReady() 中保留 IR 版本校验，模型不合格时降级为能量门控而不是崩溃。这个教训的普遍意义是：**端侧部署里，模型文件本身也是接口，版本契约必须显式校验。**

录音保活与 10 秒检查点。 会议录音动辄几十分钟，Android 后台回收是真实威胁：进程被杀意味着整段录音报废。方案是两层：①前台服务（Foreground Service）保活，录音期间持有持久通知，系统不会因内存压力轻易回收；②每 10 秒把已采集的 PCM 与已转写文本落一次检查点，即便最坏情况发生（断电、系统强杀），损失也被限制在最后 10 秒以内。这个机制在第六章真机测试中经受了 12 分钟全负载考验。

5.2 中转服务端

FastAPI + SQLite 单文件数据库，systemd 托管自动拉起，nginx + Let's Encrypt 提供 HTTPS（https://api.ai-note.top）。部署在 2 核 1GB 的入门级云主机上，中转服务本身只做认证、计数和转发，这个配置足够，也说明这套隐私架构的运营成本极低。

第六章 系统测试与实验分析

6.0 实验平台与数据集

实验平台。 三组对照实验使用 Python 评测脚本执行（sherpa-onnx 1.13.4），加载与 App 完全相同的模型文件，逐行复现 App 的切窗与聚类算法；推理使用桌面 CPU 双线程。需要说明的是：①sherpa-onnx 的 Python 绑定不暴露 Silero VAD 类，实验一用 RMS 能量门控（25 毫秒帧、10 毫秒步长、阈值随噪声底自适应）模拟 VAD 门控，其余缓冲与切割逻辑与 App 完全一致；②桌面测得的绝对时延反映的是模型计算代价，配置间的相对结论可以平移到手机端，真机绝对表现由 6.4 节的真机测试单独覆盖。所有 CER 均在去除标点与空格后按中文字符计算。真机为 iQOO Neo 8（骁龙 8+ Gen 1，Android 16），App 端 sherpa-onnx 1.12.1。

数据集。 三个数据集（D1/D2 为自建，D3 为自建对话 + 公开会议语料混合），全部真实语音、人工标注：①D1 干净语音：32 段（每段 9~40 秒），安静房间朗读，逐字校对参考文本；②D2 噪声语音：31 段（每段 7~39 秒），街道/咖啡厅/风扇等背景噪声，人工转写；③D3 多人对话：31 段，分两个子集：11 段自建对话（conv，*绝大多数 2 人，其中 1 段实为单人），人工标注"起止时间-说话人"三元组；20 段公开会议语料 AliMeeting（M2MeT 挑战赛 [15]，CC-BY-NC-4.0 许可，仅用于学术研究）测试集切片（ali*），12 段 3 人 + 8 段 4 人远场真实会议，每段 90~150 秒，取 8 通道录音的第 1 通道，官方 RTTM 标注转换为同一格式。D3 因此覆盖真实多人会议，不再只有自建双人对话。

6.1 实验一：切窗参数对识别效果的影响

实验目的。 切窗策略有两个旋钮：预滚缓冲长度与停顿切窗阈值。切太急会吃掉句首字、切碎句子；等太久字幕出得慢。本实验用网格搜索给这两个旋钮找最佳位置，并验证核心假设：切窗不损失整段识别的精度。

实验设置。 数据集 D1（32 段），int8 模型，最大段长放宽到 30 秒（比 App 的 10~15 秒宽，隔离停顿阈值的效应）。扫描预滚 ∈ {0, 0.3, 0.6, 1.0s} × 停顿阈值 ∈ {0.4, 0.6, 1.0s} 共 12 组。指标：①CER，全部段识别结果拼接后与参考文本的字错率；②句首缺字率，全局字符对齐后，段首字对齐位置落后于期望起点的段占比，专门捕捉"起判滞后吃字"；③平均延迟估计=停顿检测等待 + 实测平均单段解码耗时。

实验结果。

预滚(s)	停顿阈值(s)	CER(%)	句首缺字率(%)	延迟(s)
0	0.4	4.17	2.60	1.65
0	0.6	3.47	4.69	3.51
0	1.0	3.22	0.00	4.94
0.3	0.4	3.48	1.56	1.77
0.3	0.6	3.37	4.69	3.60
0.3	1.0	3.22	0.00	4.94
0.6	0.4	3.56	1.56	1.86
0.6	0.6	3.23	3.12	3.55
0.6	**1.0**	**2.98**	**0.00**	5.16
1.0	0.4	3.66	1.56	1.86
1.0	0.6	3.25	3.12	3.71
1.0	1.0	3.04	0.00	5.13

结果分析。 三点结论。①**切窗不伤精度**：12 组配置的 CER 全部落在 2.98%~4.17% 的窄带内，证明"VAD 门控 + 预滚 + 双阈值"的切法保住了整段识别的精度底线，这是整个策略成立的前提。②**两个旋钮各管一件事**：停顿阈值主导延迟与缺字：1.0 秒阈值下任何预滚的缺字率都是零，但延迟被推到约 5 秒；0.4 秒阈值把延迟压到 1.7~1.9 秒，代价是最多约 1 个百分点的 CER。预滚负责在固定阈值下"捡回"精度：0.6 秒停顿档上，预滚从 0 加到 0.6 秒，CER 从 3.47% 降到 3.23%，缺字率从 4.69% 降到 3.12%；再加到 1.0 秒则毫无收益（3.25%/3.12%）。③**0.6 秒预滚是收益饱和点**：它远超 VAD 实测起判滞后（约 0.25 秒以内），留了一倍以上余量，又不白白拉长每段音频。工程结论：App 默认 0.6 秒预滚 + 0.6 秒停顿是精度与延迟的平衡点（CER 3.23%、延迟 3.55 秒）；对延迟不敏感的场景（如讲座归档）可切到 1.0 秒停顿，换取绝对最优的 2.98% 与零缺字。0.6 秒停顿档残余的 3.12% 缺字来自极短插入语（"嗯""好"），其首音能量低于门控阈值，预滚也救不回——这是该档位的固有代价。

6.2 实验二：在线说话人聚类

实验目的。 回答三件事：①在扩充后的 D3 上，在线聚类算法各组件各贡献多少（消融）；②变体B 的两个超参数（前瞻窗口 N、切换惩罚 δ）取多少才算稳健；③两个工程选型问题：公开基准上更强的嵌入模型 ERes2Net 值不值得换、会后离线精修还能再捡回多少精度。

实验设置。 数据集 D3（31 段，构成见 6.0 节）。四个在线变体共享同一套 VAD 切窗与 CAM++ 嵌入（算法细节见 4.2 节）：基线为固定阈值 0.6；变体A 加入时长感知规则与连续性先验；变体B 在 A 之上加入有界延迟重判（N=3、δ=0.05）；自适应阈值分别与变体A、变体B 组合成两组，检验其单独与组合效果。原实验的离线基线（自实现均匀窗方案，非 pyannote：均匀 2 秒窗 + CAM++ 嵌入 + 录音结束后层次聚类）在双人子集上保留作对照。

评估口径。 逐文件计算：预测簇与真实说话人经匈牙利算法最优匹配后，按段时长加权得到该文件的归属准确率；本节所有汇总数字均为 31 个文件的逐文件加权准确率的算术平均（涉及子集时分别给出 conv/ali 均值）。

实验结果（消融）。

变体	加权准确率(%)	平均预测簇数	相对基线(pp)
基线（固定阈值）	68.8	7.1	—
A：时长感知+连续性	71.9	4.0	+3.0
B：A+有界延迟重判（最终在线配置）	**76.6**	**3.6**	**+7.8**
A+自适应阈值	71.7	4.1	+2.8
B+自适应阈值（组件分析，非最终）	77.4	2.9	+8.6

结果分析。 ①**固定阈值基线严重过聚类**：真实人数 1~4 人，它平均预测 7.1 个簇，原因是短段与跨话轮边界段不断开出伪簇。②**时长感知规则是第一大收益**：平均簇数压到 4.0，准确率 +3.0pp。③**有界延迟重判是第二大收益**：把边界附近的误判用动态规划重新裁决，准确率升到 76.6%，平均簇数 3.6 已贴近真实人数，代价是标注固定滞后 3 段。④**最终在线配置为变体B 固定阈值 0.6（无自适应）**：76.6%（conv 76.6%、ali 76.5%），比固定阈值基线高 7.8 个百分点，两个子集收益一致，说明方法从自建双人对话迁移到真实远场会议没有失效。自适应阈值与变体B 叠加可再得 +0.9pp（77.4%，conv 77.1%、ali 77.6%），其主要作用不是改判而是抑制碎簇（平均预测簇数从 3.6 降到 2.9），但逐文件看增益并非单调：多段明显提升的同时有两段明显变差（ali13 −21.9pp、ali12 −14.8pp）。考虑到增益微弱、逐文件不均且引入额外的状态依赖，本文不采纳该组件，将其作为被实验否定的微增组件保留分析（幸存者偏差分析见本节末）。⑤与旧实验离线基线的对比仍然成立：双人子集上在线法 72.6% 对 46.3%（11 段 conv 均值，总耗时 5.6 秒对 6.8 秒）：2 秒均匀窗声纹信息不足且常横

跨话轮边界，离线法对双人对话预测出 7~12 个簇。需要如实说明：该离线基线是"均匀窗+层次聚类"（自实现，非 pyannote）这一能在同等移动端约束下运行的方案，并非完整 pyannote 管线；换用带分割模型的离线系统，准确率可能追平甚至反超，但"必须等录音结束"与"跑不进手机单线程执行器"两条结论不受影响。另需说明：评测脚本只保留 ≥1.0 秒的段，不足 1 秒的继承规则在本评测中极少触发，它在真机 VAD 短段更多的场景里作用更大。

超参数敏感性。 变体B 在 N ∈ {3,5,7} × δ ∈ {0.03,0.05,0.08,0.12} 的 12 组网格上扫描（整体加权准确率，%）：

N \ δ	0.03	0.05	0.08	0.12
3	74.6	**76.6**	76.4	75.1
5	74.6	**76.6**	76.4	75.5
7	74.6	**76.6**	76.4	75.5

δ=0.05 最优且处于平台区——δ=0.08 仅低 0.17pp，选择不脆弱；N≥3 即饱和，故选标注延迟最低的 N=3。

嵌入模型选型（负结果）。 用公开说话人确认基准上更强的 ERes2Net-base（512 维）挑战 CAM++，变体B 固定 N=3、δ=0.05：

嵌入模型	加权准确率(%)	单段嵌入耗时(ms)	嵌入 RTF
CAM++（192 维）	**76.6**	**238**	**0.044**
ERes2Net-base（512 维）	74.3	539	0.101

挑战双线失败：准确率反而低 2.3 个百分点，单段嵌入慢 2.3 倍。公开基准的高分没有迁移到"VAD 话轮段 + 在线质心"的会议场景，而更慢在手机单线程执行器上是实打实的代价。CAM++ 保留。这个负结果就是选型依据本身。

会后离线精修。 以在线结果为初始划分，用缓存嵌入重聚（算法见 4.2 节组件④）。最终配置以变体B 固定阈值在线划分（76.6%）初始化，并含质心相似簇合并（T=0.6，与在线阈值同源）；两个独立实验复现了下表完全相同的结果。

方法	加权准确率(%)	平均簇数	精修耗时
在线变体B（固定阈值，最终在线配置）	76.6	3.6	—
+会后精修（最终配置）	**79.7**	3.6	单段均值 0.021s

嵌入在录音期间已在线算好并缓存，精修只付聚类的钱：单段均值 0.021 秒再捡回 3.1 个百分点，两个子集一致（conv 76.6%→79.9%、ali 76.5%→79.6%）。**诚实短板**：在线阶段有 21/31 段的预测簇数不等于真实人数（平均绝对误差 1.10 人），精修后仍有 14/31 段（平均绝对误差 1.00 人）：轮廓系数选 k 对归属的改善远比对计数可靠，本文不声称实现了可靠的自动人数识别。

自适应阈值：幸存者偏差分析。 自适应阈值（最近 50 个已接受相似度，clamp(μ−1.5σ, 0.45, 0.75)，前 6 个样本冷启动用 0.6）的初衷是让阈值跟随每个会议的声学条件。实验否定了它的独立价值：变体A+自适应 71.7%，并不比变体A 单独的 71.9% 好。逐样本排查发现原因是幸存者偏差：分布统计只覆盖**已接受**的相似度样本，它们集中在高位，据此拟合的阈值只会上调；而真正制造错误的样本（相似度落在接受带以下的短段）从未进入已接受集合，也就无法把阈值拉下来。阈值恰好在该放松的地方收紧。但与变体B 组合后，自适应阈值表面上挣到了一点分数：B+自适应整体 77.4%（conv 77.1%、ali 77.6%），比固定阈值的变体B 再高 +0.9pp；平均预测簇数从 3.6 压到 2.9；组合语境下它的主要作用是在 DP 重判内部抑制碎簇发射（阈值同时作用于新簇发射代价

与归入判定）。然而收益并非逐文件单调（ali13、ali12 分别变差 21.9pp 和 14.8pp），+0.9pp 的聚合提升在 31 段样本上并不稳健，且该组件引入运行分布状态、增加冷启动与调参面。综合权衡，最终配置不采纳自适应阈值，保留固定阈值 0.6；它作为一次"聚合微增、逐文件有回退、机制上存在幸存者偏差"的被否定组件如实报告，而不是免费的午餐。

局限声明与失败尝试。 ①说话人数量估计不可靠（精修后仍有 14/31 段计数错误，平均绝对误差 1.00 人，见上）；②全部实验在单一桌面平台执行，真机验证只有一款机型，绝对耗时随设备而异；③交叠语音未解决：多人同时说话会同时干扰识别与嵌入，所有变体都未建模交叠；④D1/D2 为单人自建朗读集（32/31 段），规模有限——D3 已引入 AliMeeting 公开语料，D1/D2 的公开语料扩标列为未来工作。两个失败尝试如实记录，为第三点划定边界：多尺度重嵌入二次精修（相邻同簇段拼接至 ≤10 秒重提嵌入再聚类）准确率反降 1.5pp 且每条会话耗时 4.3 秒；拼接段一旦混入误归属段，更长的嵌入反而放大错误；用 pyannote 分割模型剔除交叠帧后再提嵌入，整体同样反降 2.0pp；高交叠文件剔除后存活语音太稀疏（ali15 交叠帧占 40% 时嵌入质量崩溃），交叠处理需要比丢帧更细粒度的方案。对应未来工作：把学习型分割前端用于交叠感知归属而非删帧（如 pyannote.segmentation [6]）、s-norm 分数归一化、显式交叠建模。

6.3 实验三：模型配置的性能-精度权衡

实验目的。 为 App 选默认配置。需要回答三个问题：int8 的精度是否够用，fp32 的 936MB 体积是否值得，以及 GTCRN 降噪应否默认开启。

实验设置。 D1+D2 共 63 条音频，每条用三种配置各识别一遍：int8 / fp32 / int8+GTCRN 降噪。指标：CER、RTF（解码耗时÷音频时长）、峰值内存。

实验结果。

配置	CER 干净(D1, %)	CER 噪声(D2, %)	总平均 CER(%)	平均 RTF	模型体积
int8	2.71	3.77	**3.23**	**0.344**	**230 MB**
fp32	**2.54**	4.06	3.28	0.380	936 MB
int8+GTCRN	2.81	3.91	3.36	0.599	230+0.5 MB

结果分析。 ①**int8 精度中性**：总平均 CER 3.23% 对 3.28%，干净子集 fp32 略好、噪声子集 int8 反而略好，差异在两个方向都出现，说明处在逐条波动范围内，量化没有造成实质损失。同时 int8 快 10%（RTF 0.344 对 0.380）、体积只有四分之一。移动端默认选 int8，是有实验证据的工程决策。②**GTCRN 在本数据上得不偿失**：CER 两个子集都略微变差，RTF 却涨了 74%（0.599）：降噪一遍的耗时接近识别本身。原因是 D2 的噪声强度中等，SenseVoice 训练数据本就含大量真实噪声，对此强度已有鲁棒性，降噪的"干净化"反而抹掉了有用的声学细节。工程结论：降噪保留为恶劣声学环境的可选开关，不作默认。③两点测量说明：RTF 测自桌面 CPU 双线程，仅刻画模型计算量，App 是单线程执行器，手机端绝对 RTF 会更高，但配置间的相对次序不变（真机侧，6.5 节的 66 分钟连续全负载运行无积压完成，意味着端到端 RTF < 1）；峰值进程内存在三种配置下几乎相同（约 1.8GB），因为同一长驻进程被 ONNX Runtime 内存池主导，故以模型文件体积作为内存差异的有效度量。

大型模型参照（Paraformer-large）。 为标定端侧小模型与大模型的距离，额外用 Paraformer-large（int8）在与实验三完全相同的切窗流水线上解码 D1、D2：

模型	CER 干净(D1, %)	CER 噪声(D2, %)	RTF (D1 / D2)
Paraformer-large int8	3.12	11.2	0.286 / 0.267
SenseVoice-Small int8	**2.71**	**3.77**	0.342 / 0.346
SenseVoice-Small fp32	2.54	4.06	0.384 / 0.376

干净语音上端侧小模型与大模型基本打平（2.71% 对 3.12%，fp32 为 2.54%）。噪声集的差距（3.77% 对 11.2%）很大，但存在混杂因素，两种口径都如实交代：同一套流水线口径下，能量门控在噪声音频上过度切段，而 Paraformer 对碎片段远比 SenseVoice 敏感：最差的 d2_18 改为整段直解码后 CER 从 46.3% 降到 0.0%（另两段：30.9%→7.3%、34.9%→16.3%）。因此 11.2% 是"同一条切窗流水线"口径下的数字，不是 Paraformer 的真实水平。诚实的结论是两条：①干净语音上，量化小模型相对数倍体积的大模型没有损失；②SenseVoice 在噪声下对本切窗器的鲁棒性本身，就是选它作端侧默认的理由之一。Paraformer 在本评测中的峰值内存约 0.42~0.45GB RSS。

6.4 真机稳定性与耗电测试

在 iQOO Neo 8 上进行 12 分钟连续会议模式录音，实时转写、在线说话人聚类、AI 纪要生成全部开启，外放语音模拟真实会议。结果：①无崩溃、无丢音，实时字幕全程流畅；②进程内存每 30 秒采样，从 565MB 缓增至 604MB（+6.9%）：浅而近似线性的增长来自录音与转写缓冲本身变长，不是泄漏（泄漏的特征是与业务数据量无关的无界增长）；③电量计前后读数均为 34%，即全音频管线 12 分钟总耗电低于电量计 1% 的分辨率。60 分钟耐久测试见 6.5 节。

6.5 耐力与稳定性测试（60 分钟全负载）

在 iQOO Neo 8（骁龙 8+ Gen 1，Android 16，debug 包 v1.3）上进行 66 分钟耐力测试。测试方法如实说明：将一段 66 分钟 AliMeeting 真实会议音频经 PC 扬声器播放，手机在旁以会议模式连续录制转写，这是一条声学复读链路，仅用作稳定性负载，不用于精度评估。运行期间每分钟采样一次电量、电池温度、进程 RSS 与 CPU。

结果。 连续录音转写 65 分 33 秒，零崩溃零重启；音频完整落盘 126.57MB（理论值 126.7MB），无丢失。电量从 56% 降到录音停止时的 53%（全程净耗 3%，电量计分辨率 1%）。电池温度从 37.6°C 升至录音末段的 41.6°C（录音期峰值 43.0°C），停止后数分钟内回落到约 41°C。进程 RSS 在模型加载完成后进入约 610MB 的平台，随后以递减的增速缓升至停止前约 0.84~0.90GB（这是录音、转写与嵌入状态的自然累积，而非失控泄漏），管线空闲后回落至约 440~600MB。逐分钟日志中出现过一次 3.2GB 的孤立 RSS 采样点（停止前一分钟，下一采样即恢复正常），它未伴随崩溃、丢音或可见卡顿，我们将其判读为瞬时统计（如 GC 时点的账面值）而非稳定性缺陷，但如实报告。

诚实的适用范围。 这是单机型、复读链路测试：它证明的是全管线在主流旗舰机上扛得住一小时级会议；多机型、真人多人现场的耐力与精度联合评估留作后续工作。

6.6 实验六（初步）：端侧会议纪要小模型的领域适配与迭代

实验目的。 第四章把端侧大模型摘要列为未来工作，本实验把它变成已做工作。要回答三个问题：①"真实会议蒸馏、合成扩充、负例边界"的数据配方，能否让 1.7B 级小模型学会"带说话人标注的会议转写到纪要"这一领域任务；②"缺陷发现、定向扩数据、复训验证"的迭代闭环是否有效；③适配后的模型量化到手机可承载的体积后还能否工作。

数据集。 自建会议摘要微调数据集，三层构成见下表。第一层取 6.0 节 D3 的 AliMeeting 切片（ali01-20），用端侧同款 SenseVoice-Small int8 逐段识别、按说话人合并成"说话人N：…"格式转写，保留 ASR 真实噪声不做人工清洗，刻意模拟端侧真实输入分布；摘要由 DeepSeek（deepseek-chat）按"忠实原文、禁止编造"的要求生成，每份转写出两种任务（150 字内简要纪要、纪要加待办清单）。第二层由 DeepSeek 合成 16 类场景（学习小组、项目评审、家庭会议、客服协调等）的会议，单次调用同时产出转写与两种摘要，并要求带口语化插话、跑题、打断。第三层从第一层派生负例：截短、全员改标"说话人1"的独白、字符级扰动各 10 条，训练模型在信息不足时如实说明而不是硬写。一轮训练集 350 条、评估集 30 条；第二轮针对一轮暴露的三个缺陷定向生成 161 条（全部合成，不含 AliMeeting 派生物），配 15 条定向评估集，三个靶点各 5 条，与训练集不混。二轮全部样本经过数字逐字溯源校验。

层	来源	训练条数	设计意图
1 真实会议蒸馏	AliMeeting 切片 + 端侧 SenseVoice 转写 + DeepSeek 摘要	20（10 会议 × 2 任务）	锚定真实输入分布（含 ASR 噪声）
2 合成扩充	DeepSeek 生成 16 类场景	300（150 会议 × 2 任务）	扩场景覆盖，带标记可与真实区分
3 负例与边界	第 1 层派生：截短 / 独白 / 扰动各 10	30	教"信息不足就直说"
二轮定向补充	全部合成，三个靶点	161	无待办 49 + 不定负责人 48 + 数字密集 64

许可证声明：AliMeeting 音频与标注为 CC-BY-NC-4.0 许可（仅学术研究、禁止商用）[15]，派生的转写与摘要同样受此约束；DeepSeek 蒸馏数据仅用于本研究，数据集若对外发布，需先复核 DeepSeek 服务条款对蒸馏与商用的限制。

训练与量化。 基座模型 Qwen3-1.7B [20]，LoRA 秩 16 [19]，ms-swift 框架 [21]，单张 A10。一轮 350 条 × 3 轮共 66 步，用时 8 分 55 秒，训练 loss 从 3.08 收敛到 1.09；二轮 511 条（350+161）× 3 轮共 96 步，终值 loss 1.04。部署侧：LoRA 合并回基座得 bf16 模型 3.3GB，经 llama.cpp [22] 转 GGUF 并量化到 Q4_K_M，得 1.03GiB，比 f16 转换小 68%；桌面 CPU 冒烟测试用真实会议转写跑通纪要生成，5.7 tokens/s，输出格式合格。手机端的推理速度、内存与耗电尚未实测，属于 App 集成 llama.cpp（JNI）之后的工作。

一轮质检：三个缺陷与一个数据真相。 一轮模型在评估样本上暴露三类缺陷，均有原始输出存档：①编造待办，纯分享型会议被脑补出行动项，样本 r2a050 的原文开场就声明"不布置任务"，模型仍编出 4 条待办并逐条分给说话人 1~4；②编造负责人，原文说"回头群里认领"，模型却把待办安到具体说话人头上；③数字漂移，电费 240 元写成 250 元，次要数字漏写。缺陷①的根因随后在数据里被找到：一轮 350 条训练样本中，"无待办"样本是 0 条。模型脑补待办不是笨，是数据从没教过它"可以没有待办"。模型缺陷与数据缺陷互为印证，这是本实验数据工程的核心发现：领域小模型的行为问题，先回数据里找答案。

二轮定向修复与三靶对比。 第二轮按靶造数：靶 A 生成"纯讨论、无任何行动项"的会议，服务器端强制校验待办段以"无"开头；靶 B 生成"事项明确但全程不定人"的会议，校验每条待办标"（负责人：待定）"且不点名；靶 C 生成数字密集会议（预算、排班、报价、盘点），单轮生成漂移率过高（32 条只过 5 条），改两步法（先生成转写、低温度单独摘要）加服务器端数字逐字校验，漂移样本直接丢弃。复训后两版模型在 15 条定向评估集上人工审读对比（口径：token 级保真率为输出中全部阿拉伯数字串可在转写原文逐字溯源的比例；样本级全保真率为整篇纪要零不可溯源数字的样本占比；三靶各 5 条，人工判定）：

指标（每靶 n=5）	v1（350 条）	v2（511 条）
靶 A：无待办场景不编造待办	20%	80%
靶 B：不定人场景不编造负责人	20%	80%
靶 C：数字 token 级保真率	94%	98%（119/121）
靶 C：数字样本级全保真率	80%	60%

注：每靶 n=5，样本级比例的 Clopper–Pearson 95% 精确区间为 20%→[0.5%, 71.6%]、80%→[28.4%, 99.5%]；小样本下方向可信、精度数字待扩样确认。

两升一降，都如实报告。靶 A 上 v2 有 4/5 正确输出"待办：无"：同一个 r2a050，v1 在纪要正文承认"未布置任务"之后仍编出 4 条待办，v2 则直接给出"待办：无"。靶 B 上 v2 学会把"回头群里定"翻译成"（负责人：待定）"。靶 C 的样本级指标从 80% 降到 60%，唯一翻车样本是 r2c023：原文把电话号码拆在两处说（"138 开头""尾号 7421"），v2 把它缝合成完整号码 1387421 写进待办。这不是凭空编造，是部分信息拼接错误，但对用户同样是误导，下一轮方向就是号码、日期类拼接负例。

诚实边界。 ①每靶仅 5 条评估样本，n=5 的对比只够证明方向可信，精度数字需扩样后确认；②小模型纪要质量不及云端 DeepSeek，定位是离线草稿，与云端精修构成免费与付费分层，不是替代；③1.03GiB 体积与桌面 CPU 冒烟只证明"能跑"，真机表现待集成后实测；④蒸馏数据与 AliMeeting 派生物的许可证限制见上文，这决定了该数据集与适配模型只能留在研究语境。

第七章 总结与展望

7.1 工作总结

本文针对云端语音识别笔记应用的隐私风险，设计并实现了端云边界的隐私保护智能笔记系统 AI Note，主要完成四项工作：①端云边界架构：音频处理全部端侧完成，文本按需经带认证、配额、限流、白名单的自建中转服务上云，直接呼应《个人信息保护法》对敏感个人信息"最小必要"的要求；②流式化切窗策略：实验证明其在整个参数网格上保持整段识别精度（CER 2.98%~4.17%），0.6 秒预滚 + 1.0 秒停顿取得最优 2.98% 与零句首缺字；③在线说话人聚类：31 段对话（含 20 段 AliMeeting 真实 3~4 人会议）上在线 76.6%、会后毫秒级精修 79.7%，比固定阈值基线高 7.8 个百分点，并大幅超过同约束离线基线（双人子集 46.3%），且录音停止零等待；④会议纪要小模型领域适配：三层数据配方加一轮缺陷驱动的定向迭代，编造待办与编造负责人的不犯率从 20% 提升至 80%，数字 token 级保真率 98%，量化后 1.03GB 的模型在纯 CPU 上跑通真实会议纪要生成，端侧大模型摘要从未来工作变成已验证的路径（草稿级质量与每靶 5 条的样本量边界见 6.6 节）。配合 int8 量化中性（3.23% 对 3.28%，体积 1/4）与 66 分钟真机全负载耐力的实测，可以回答绪论提出的问题：全端侧的隐私保护智能笔记，在今天的手机上是可行的。

7.2 不足与展望

①重叠语音：多人同时说话时，单通道识别与嵌入都会混淆，需要重叠感知分割，可引入 pyannote 分割前端与 s-norm 分数归一化；②说话人数量估计不可靠（精修后仍有 14/31 段簇数不等于真实人数，平均绝对误差 1.00 人），且基于已接受样本的自适应阈值存在幸存者偏差（见 6.2 节），需要能触及被拒绝样本的阈值机制；本文离线对照为自实现的均匀窗方案，与完整 pyannote 管线的 DER 对照实验列为未来工作；③文本级隐私缺口已有阶段性答案：1.03GB 的端侧纪要模型可以离线出草稿（6.6 节），但质量仍不及云端，手机端速度、内存、耗电尚未实测；下一步是 App 集成 llama.cpp 后的真机实测、针对数字拼接错误的第三轮负例扩样，以及把每靶

5 条的定向评估集扩到能下精度结论的规模；④其他：多机型与真人多人现场的耐力-精度联合评估（6.5 节为单机型复读链路结果）、领域热词定制。

成稿后的版本说明。 本文所述系统与实验基于 2026 年 8 月上旬的 v1.0 架构快照。截至发稿，系统已迭代至 v1.4：新增第二识别引擎（Paraformer）、DFN3 极致降噪、FireRedASR 离线精修、知识星图可视化，并将第六章的端侧纪要路径产品化——离线纪要与会议实时离线摘要已随版本上线。上述迭代不改变本文的架构设计与实验结论。

AI 辅助声明

本文写作过程中使用了 AI 助手（大语言模型工具）辅助语言润色与部分文字的起草。本文的系统设计与实现、实验设计、数据采集与标注、全部实验测量与结果均由作者本人完成。作者已审阅、核实并修改全部 AI 辅助内容，对本文的完整性与准确性承担全部责任。

参考文献

[1] Radford A, Kim J W, Xu T, et al. Robust speech recognition via large-scale weak supervision[C]//Proc. ICML, 2023. [2] Gao Z, Zhang S, McLoughlin I, et al. Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition[C]//Proc. Interspeech, 2022. [3] FunAudioLLM. SenseVoice: Multilingual voice understanding model[EB/OL]. 2024. https://github.com/FunAudioLLM/SenseVoice [4] Next-gen Kaldi. sherpa-onnx: Speech-to-text, text-to-speech, and speaker recognition using onnxruntime[EB/OL]. 2022–2025. https://github.com/k2-fsa/sherpa-onnx [5] Silero Team. Silero VAD: pre-trained enterprise-grade voice activity detector[EB/OL]. 2021. https://github.com/snakers4/silero-vad [6] Bredin H, Laurent A. End-to-end speaker segmentation for overlap-aware resegmentation[C]//Proc. Interspeech, 2021. [7] Wang H, et al. CAM++: A fast and efficient network for speaker verification using context-aware masking[C]//Proc. Interspeech, 2023. [8] Zhang A, Wang Q, Zhu Z, et al. Fully supervised speaker diarization[C]//Proc. ICASSP, 2019. [9] Dimitriadis D, Fousek P. Developing on-line speaker diarization system[C]//Proc. Interspeech, 2017. [10] Rong X, Sun T, Zhang X, et al. GTCRN: A speech enhancement model requiring ultralow computational resources[C]//Proc. ICASSP, 2024: 971-975. [11] Vaswani A, et al. Attention is all you need[C]//Proc. NeurIPS, 2017. [12] DeepSeek-AI. DeepSeek-V3 technical report[R]. 2024. [13] 全国人民代表大会常务委员会. 中华人民共和国个人信息保护法[Z]. 2021. [14] 全国人民代表大会常务委员会. 中华人民共和国数据安全法[Z]. 2021. [15] Yu F, Zhang S, Fu Y, et al. M2MeT: The ICASSP 2022 multi-channel multi-party meeting transcription challenge[C]//Proc. ICASSP, 2022. [16] cactus-compute. needle: 45M 参数端侧 tool-calling 模型[EB/OL]. 2026. https://github.com/cactus-compute/needle [17] Google. AI Edge Gallery: 端侧生成式 AI 参考应用[EB/OL]. 2025. https://github.com/google-ai-edge/gallery [18] altic-dev. FluidVoice: macOS 端侧听写产品[EB/OL]. 2025. https://github.com/altic-dev/FluidVoice [19] Hu E J, Shen Y, Wallis P, et al. LoRA: Low-rank adaptation of large language models[C]//Proc. ICLR, 2022. [20] Yang A, Li A, Yang B, et al. Qwen3 technical report[R]. arXiv:2505.09388, 2025. [21] ModelScope. ms-swift: 大模型轻量微调与推理框架[EB/OL]. 2023. https://github.com/modelscope/ms-swift [22]

Gerganov G, et al. llama.cpp: LLM inference in C/C++[EB/OL]. 2023. https://github.com/ggml-org/llama.cpp