arXiv:2603.08397v1 [eess.AS] 9 Mar 2026

NLE：通过转录文本编辑实现的基于大语言模型的非自回归自动语音识别

Avihu Dekel、*Samuel Thomas*、*Takashi Fukada*、*George Saon*

IBM
Researc
havihu.dekel@ibm.com

摘要

尽管基于大语言模型的自回归自动语音识别系统准确率很高，但其顺序解码限制了并行性，并导致延迟较高。我们提出 NLE，这是一种非自回归方法，将语音识别建模为条件转录文本编辑，从而实现完全并行的预测。NLE 从预训练语音编码器中提取声学嵌入和初始假设，然后使用通过潜在对齐目标训练的双向大语言模型编辑器来优化该假设。交错填充策略利用 Transformer 的恒等映射偏置，使模型能够专注于修正，而不是从头重建。在 Open ASR 排行榜上，NLE++ 的平均词错误率为 5.67%，实时因子（RTFx，即实时因子的倒数）为 1630。在单句场景中，NLE 的速度比自回归基线快 27 倍，适用于实时应用。

索引词：语音识别、文本编辑、非自回归

1. 引言

基于大语言模型的自动语音识别系统 [1, 2] 将预训练语音编码器与预训练大语言模型配对，通常通过学习型投影器将声学表示映射到大语言模型嵌入空间，从而提高转录准确率。在大多数现有系统中，大语言模型充当自回归解码器，每次生成一个令牌。尽管这种设计能够取得较高的准确率，但自回归解码的序列特性限制了并行性，带来了显著的端到端延迟，并导致推理速度变慢（实时因子较低；实时因子衡量音频时长与处理时间之比）。在无法进行批处理的实时对话场景中，这一局限尤为严重，因为无法并行执行令牌生成会直接导致较高的单话语延迟。此外，这些系统会舍弃语音编码器通常生成的初始假设，尽管该假设往往能提供合理的草稿，可在此基础上进行润色，而非从头重新生成。本文致力于解决这些局限，并实现可并行化的基于大语言模型的推理。

连接主义时序分类 (CTC) [3] 为将较长的声学序列转换为更短的词元序列。CTC 编码器会生成帧级词元后验概率，而推理过程则通过高效、完全并行的解码流程合并重复词元并移除空白符。不过，CTC解码受到条件独立性和单调对齐假设的限制。此外，CTC模型的语言建模能力有限，也缺乏大型预训练语言模型所具备的广泛语言先验，因此当声学证据较弱时，难以恢复合理的内容。在实际应用中，CTC

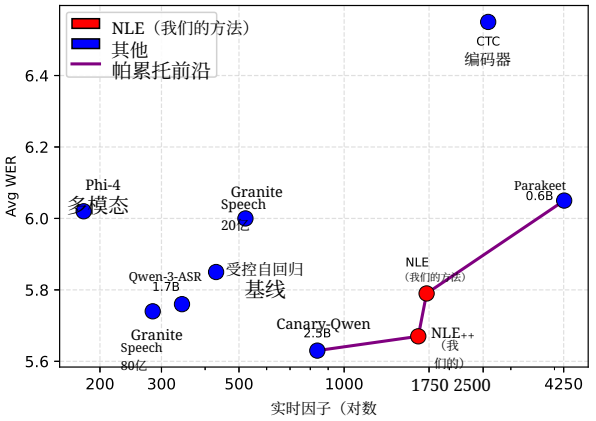


图 1：*Open ASR* 排行榜中 *NLE* 与 *NLE++* 相对于排名前六模型的词错误率（*WER*）—实时因子（*RTFx*）权衡（截至 *2026*年*2*月）。*NLE* 的两个变体均位于帕累托前沿上（没有其他模型能够同时实现更低的 *WER* 和更高的 *RTFx*），在保持具有竞争力的准确率的同时，实现了更高的推理速度。

输出往往会出现局部错误，例如音素替换，或词语删除等结构性错误，在噪声较大或存在歧义的情况下尤其如此。许多此类错误具有系统性，且可在局部修正，因此CTC假设很适合作为后续编辑的草稿，而非进行完整的重新生成。

我们提出了 **NLE**，一种基于大语言模型的**非**自回归**大**语言模型**编**辑自动语音识别系统。NLE 将基于大语言模型的语音识别重新定义为条件转录文本编辑：NLE 不再以自回归方式解码词元，而是编辑从预训练语音编码器中提取的假设文本，并利用来自同一编码器的声学上下文进行引导。这种编辑形式支持完全并行的预测，从而实现快速推理。

我们的方法将预训练大语言模型适配到编辑任务，以利用其语言知识。我们将大语言模型中的注意力机制从因果注意力修改为双向注意力 [4, 5], , 从而支持基于完整上下文的非自回归编辑。模型采用带有潜在对齐 [6], 的 CTC 风格目标进行训练，能够自然处理输入与目标转录文本之间的可变长度映射。我们使用轻量级 LoRA 适配器 [7]，使模型适应双向注意力和编辑目标。可以禁用 LoRA 适配器，并将注意力机制重新配置回因果模式，从而恢复原始大语言模型的功能。这一设计使大语言模型权重能够在自动语音识别与下游任务之间共享，例如基于转录文本的问答——这是语音增强大语言模型的一个关键应用场景 [2]。

我们采用带有显式插入 tion

槽位的交错令牌布局，以最少的词元移动实现局部插入。这种设计利用了 Transformer 的恒等映射偏置——即通过残差连接和绑定嵌入原样复制输入词元的倾向（见第 3.3 节）——使模型能够专注于纠正，而非进行完整的重构。

在 Open ASR 排行榜 [8], 这一包含 39 个强大自动语音识别模型的高难度基准上，NLE++ 取得了 5.67% 的平均词错误率，实时因子为 1630。如图 1 所示，NLE 与 NLE++均位于词错误率-实时因子空间的帕累托前沿，相较领先模型提供了出色的准确率-速度权衡。在单语句推理中，NLE 相较自回归基线实现了 27 倍加速，展现了非自回归解码在延迟敏感型应用中的显著实际优势。

2. 相关工作

2.1. 基于大语言模型的自动语音识别

近期工作探索了通过学习式投影层 [1, 9, 10, 2]，将预训练大语言模型以语音表征为条件，从而把大语言模型集成到自动语音识别系统中。与 Whisper [11], 等编码器-解码器模型类似，这些方法将语音编码器与文本解码器配对，但利用预训练大语言模型以受益于其语言知识。借助大语言模型强大的语言先验，它们提升了转录准确率，尤其是在具有挑战性的声学条件下，或在处理罕见词和专有名词时。然而，大多数基于大语言模型的自动语音识别系统依赖自回归解码，词元按顺序生成，这固有地限制了并行性，并导致较高的推理延迟和较低的吞吐量——这是实时应用的关键瓶颈。此外，当声学证据薄弱或含糊时，自回归模型可能产生看似合理但错误的内容，并且它们会丢弃语音编码器通常产生的初始假设，尽管该假设往往能提供合理的草稿。

RNN-Transducers（RNN-T） [12] 和 Token-and-Duration Transducers（TDT） [13, 14]等替代架构可借助流式处理能力加快推理速度，但它们仍按顺序生成词元，且无法利用预训练大语言模型中蕴含的广泛语言知识。我们的工作通过将大语言模型用作非自回归编辑器来解决这些局限：该编辑器并行优化初始 CTC 假设，将非自回归解码的速度与预训练大语言模型的语言知识相结合。

2.2. 非自回归自动语音识别

非自回归（NAR）自动语音识别方法旨在通过并行预测词元来降低推理延迟。连接主义时序分类（CTC） [3] 是应用最广泛的非自回归方法，它利用动态规划对齐进行边缘化，并支持高效的并行解码。然而，CTC 模型通常缺乏强大的语言建模能力，且受条件独立假设的限制，因而难以利用语言上下文进行消歧 [15, 16]。为改进 CTC，研究者提出了多种非自回归细化方法。SoftCorrect [17] 采用受约束的 CTC 损失来纠正转录文本。Mask-predict 方法 [18, 19, 20] 和迭代细化方法 [21, 22, 23]通过多轮处理，基于部分预测进行条件化，从而缓解 CTC 的条件独立性问题。

尽管非自回归方法相比自回归模型具有显著的速度优势，但它们往往难以执行插入操作（即纠正删除错误），也难以保持长程语言一致性。许多方法依赖定长预测或掩码策略，导致插入操作难以实现，或需要进行多轮细化迭代。我们的方法在利用 CTC 高效性的同时，通过两个关键思路应对这些局限：一种插入感知交错表示，无需对整个序列进行移位即可实现局部插入；以及基于双向大语言模型的编辑，利用预训练语言知识提升上下文推理能力。

2.3. 自动语音识别纠错

用于纠正首轮自动语音识别输出的后处理方法已得到广泛研究。传统方法包括 N-best列表重打分和基于格的重打分 [24, 25, 26],，通过更完善的声学模型或语言模型重新评估识别假设。近年来，基于大语言模型的纠错方法逐渐兴起：将自动语音识别转录文本输入外部大语言模型，以进行零样本或少样本纠错 [27, 28], ，从而利用其语言知识。监督式错误纠正方法则使用错误转录文本与参考文本配对的数据训练编码器-解码器模型 [29], ，使模型能够学习纠错模式。与这些针对最终自动语音识别输出进行处理的后处理方法不同，我们的方法通过基于声学嵌入的非自回归编辑，将纠错整合到解码过程中，从而联合优化声学信息与语言信息，而非仅进行纯文本纠错。

2.4. 非自回归文本编辑与翻译

机器翻译和文本编辑中的非自回归方法与自动语音识别在概念上有相似之处，因为二者都涉及在密切相关的输入输出序列之间进行映射。早期工作 [30]通过基于词汇数的并行生成引入了非自回归翻译，而莱文斯坦变换器 [31] 则通过保留、删除和插入等显式操作对文本进行编辑。与我们的工作最相关的是，基于 CTC 的潜在对齐模型已应用于非自回归翻译 [6] ，并通过显式的复制操作 [32]扩展到了文本编辑。我们将这些潜在对齐技术应用于语音识别：以声学嵌入而非单独的文本为条件，并利用经过双向注意力适配的预训练因果大语言模型，而不是从头开始训练。

3. 方法

NLE 将输入话语输入预训练的语音编码器，生成声学嵌入和一个 CTC 转录假设。随后，基于大语言模型的非自回归编辑器通过对交错排列词元序列与插入槽位应用 CTC 风格目标，预测编辑后的转录文本。图2展示了完整流程。

3.1. 提取假设与嵌入

我们使用一个预训练的基于CTC的语音编码器，并在训练期间将其冻结。我们冻结该编码器，以保留其已充分训练的声学建模能力，因为其假设的质量会直接影响整个训练过程；我们将联合微调留待未来工作。该编码器处理输入音频，并输出帧级嵌入 $H \in \mathbb{R}^{T \times d}$，其中 T 为帧数， d 为嵌入维度。它还会生成帧级CTC逻辑值，其范围覆盖一个

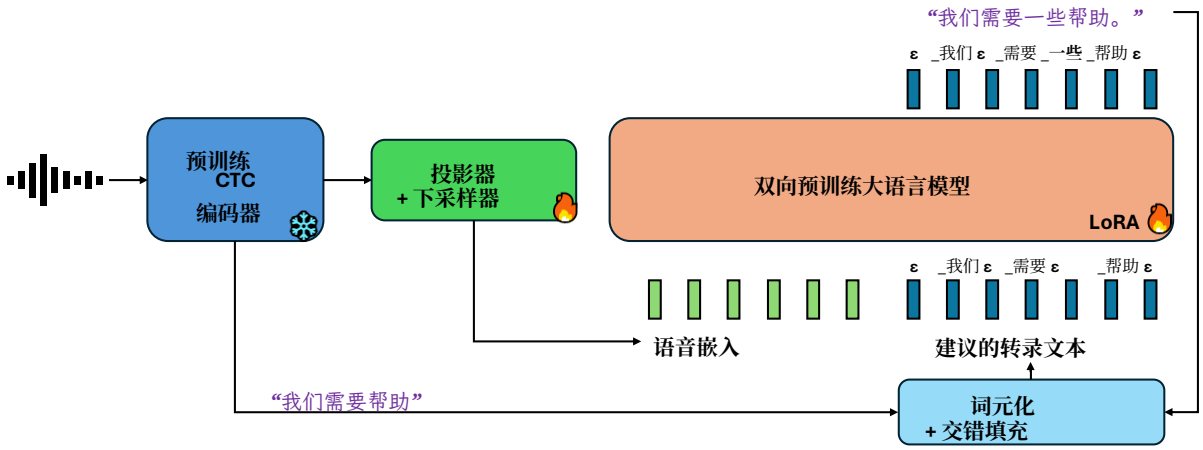


图2： *NLE* 架构概览。冻结的预训练 *CTC* 编码器会生成声学嵌入和初始 *CTC* 假设。该假设经过词元化后，与插入槽（ϵ）交错排列，再与投影后的语音嵌入拼接。经 *LoRA* 调整的双向大语言模型编辑器使用 *CTC*目标预测编辑后的转录文本。输出可以迭代地重新编辑（参见第 *5.5* 节）。

基于字符的词表，这些逻辑值通过贪心解码（最大值索引，随后进行空白移除和去重）转换为字符级字符串假设。该假设作为编辑模型的初始草稿，而嵌入 H 提供声学上下文，以指导精炼过程。

3.2. 重新分词与交错插入槽位

字符级 CTC 假设会使用大语言模型的子词分词器重新分词，以与大语言模型的词表对齐。由于大语言模型分词器是在规范文本上训练的，CTC 假设中的拼写错误（例如 “philossophy”）可能没有对应的词元，因此会被拆分为多个子词单元。我们将得到的词元序列记为 $x = (x_1, \ldots\ldots, x_N)$。

为了高效处理插入编辑，我们构建一个带有显式插入槽位的交错序列：

$$\tilde{x} = (\epsilon, x_1, \epsilon, x_2, \ldots, \epsilon, x_N, \epsilon). \quad (1)$$

其中 ϵ表示空白符号（我们复用大语言模型的序列结束标记）。这种交错表示创建了 $N+1$ 个插入槽位（每个词元之前一个，最后一个词元之后一个），从而能够进行局部插入，而无需模型移动整个下游序列。单词元插入可通过填充一个插入槽来处理。多词元插入只需在局部块内移动相邻词元以容纳新内容，序列其余部分不受影响。

为说明这一点，考虑在 x_k 和 x_{k+1}之间添加词元 (a, b, c)。给定原始序列：

$$(\ldots \epsilon, x_k, \epsilon, x_{k+1}, \epsilon, \ldots)$$

模型可通过以下逐位置预测生成此插入：将 x_k/x_{k+1} 移至左侧/右侧的 ϵ位置，并利用腾出的3个中间位置添加 (a, b, c)：

$$(\ldots, x_k, a, b, c, x_{k+1}, \ldots)$$

这可以在一次前向传播中完成，同时保留其余词元不变。插入 K 个词元需要从原始序列中更改 $2K-1$ 个词元，同时保留其余词元不变。可插入的词元最大数量为 $N+1$，在我们的训练数据中从未需要过[1]。

3.3. Transformer 中的恒等映射偏置

我们提出的编辑方法利用了 Transformer 架构固有的恒等映射偏置 [33, 34]。这种偏置源于两个关键的架构特性：

- **残差连接**能够在各层间保留输入表示，使信息得以不加改变地流经整个网络。
- **输入输出绑定嵌入**广泛用于大多数大语言模型（包括我们在 NLE 中使用的 Granite 4.0 1B Base ），并形成强烈的复制偏置：词元的嵌入与其自身在输出投影矩阵中的向量点积往往较高，这使模型自然而然地倾向于预测输入词元。

不进行编辑而直接解码交错序列时，便会自然还原出原始假设。这一特性有利于我们的任务，因为 CTC 假设中的大多数词元都是正确的，应予以保留。模型可以将学习能力集中用于识别和纠正错误，而非重构整段转录文本；这对于在实现快速并行解码的同时保持出色性能至关重要。我们还通过复制正则化目标（第 3.6 节）进一步强化这种偏置。例如，在图2中，模型只需将 $x_5 = \epsilon$ 改为“some”，其余输入词元则直接复制；由于采用了交错插入槽，无需进行词元位移。

3.4. 基于大语言模型的双向编辑器

我们使用学习型投影器P_θ 计算声学嵌入，该投影器对帧级语音嵌入 H 进行下采样，并将其映射到大语言模型嵌入空间。交错假设 $\tilde{x}$ 使用大语言模型的词元

[1]我们将输入词元的最小数量设为8，以便为短话语提供足够的插入容量。

嵌入表 E进行嵌入。这些表示沿序列维度拼接，形成大语言模型的输入：$Z = [P_\theta(H); E(\tilde{x})] \in \mathbb{R}^{(T'+2N+1)\times d_{\text{大语言模型}}}$，其中 T' 是下采样后的声学序列长度，d_{LLM} 是大语言模型的隐藏维度。

从预训练因果大语言模型出发，我们通过移除因果注意力掩码，将其注意力机制修改为双向，使每个位置都能关注所有其他位置，同时保持位置嵌入不变。这种双向上下文对于有效编辑至关重要，因为修正往往需要来自过去和未来词元的信息。我们使用 LoRA（低秩自适应）对大语言模型进行适配，从而在保留模型预训练语言知识的同时实现高效微调。LoRA 适配器可以禁用，以恢复原始因果大语言模型的功能，从而使同一套模型权重能够在自动语音识别和其他下游任务之间共享。

3.5. 基于 CTC 的编辑目标

大语言模型编辑器针对交错序列 $\tilde{x}$ 中的每个位置，在词表 $\mathcal{V}$ 上输出词元对数几率 $L \in \mathbb{R}^{(2N+1)\times|\mathcal{V}|}$。我们应用标准 CTC 损失 $\mathcal{L}_{\text{CTC}}(L, x^\star)$，通过动态规划对预测对数几率与参考转写 $x^\star$ 之间的所有有效对齐进行边缘化。
CTC目标自然地处理交错输入序列与目标转写之间的可变长度映射，使模型能够学习哪些位置应保留、删除或用于插入。这一设计选择避免了预先对齐参考文本与假设的需要，因为对齐会在损失计算过程中通过 CTC 的动态规划隐式完成。

编辑操作如下：

- **复制：** 恒等偏置使模型默认能够从输入假设中复制词元。
- **替换：** 通过在某个位置预测不同的词元来执行替换。
- **删除：** 通过在词元位置预测空白符号 ϵ 来执行删除。
- **插入：** 单词元插入使用显式插入槽。多词元插入需要移动相邻词元，模型通过 CTC 损失学习这一点；该损失允许存在多个有效对齐。

3.6. 复制正则化目标

虽然 CTC目标允许模型学习编辑操作，但并未显式施加恒等映射偏置。Transformer 层会混合不同位置的特征，而 CTC 损失允许存在多种对齐方式来生成相同的输出文本，其中包括冗余编辑（CTC 网格中的变化最终会折叠为相同的解码字符串）。在适配使用下一词元预测训练的预训练大语言模型时，这种偏置可能会被模型学到的下一词元行为覆盖。为了鼓励模型保留正确的词元，并使编辑更易于解释，我们添加了一种辅助复制正则化（CR）损失：

$$\mathcal{L}_{\text{CR}} = -\sum_{i=1}^{2N+1} \log P(\tilde{x}_i | L_i), \tag{2}$$

其中，L_i 是位置 i 处的对数几率，$\tilde{x}_i$ 是来自 $\tilde{x}$的输入词元。这种交叉熵损失促使模型预测每个位置上的输入词元，从而强化复制偏置。总训练目标变为：

$$\mathcal{L} = \mathcal{L}_{\text{CTC}} + \lambda \mathcal{L}_{\text{CR}}, \tag{3}$$

其中，λ控制正则化强度。由于 CR损失会主动鼓励模型精确预测输入序列，因此必须确保 λ 远小于 CTC 损失，以免抑制必要的编辑。

3.7. 推理

推理时，我们从冻结的语音编码器中提取声学嵌入 H 和 CTC 假设。我们对该假设重新进行词元化，并将其与空白符号交错排列，形成 $\tilde{x}$，然后将其与投影后的声学嵌入拼接，并输入双向大语言模型编辑器。编辑器会为交错序列中的每个位置生成输出对数几率 L。我们应用 CTC 贪心解码（先取最大值索引，再进行空白移除和去重），得到最终编辑后的转录文本$\hat{x}$。

关键在于，所有预测都在整个序列上并行生成，因此与自回归解码相比，推理速度显著更快。计算成本主要来自大语言模型的一次前向传播，它会同时处理所有位置。也可以将$\hat{x}$ 作为输入反馈，以迭代编辑过程并进行额外的细化，但这需要额外的前向传播（分析见第 5.5 节）。

4. 实验

4.1. 训练

模型架构。 CTC 编码器包含 4.4 亿个参数，基于 16 层 Conformer 架构 [35]，采用块大小为 200 帧的块注意力（在 50 Hz 帧率下相当于 4 秒）。该编码器处理 16 k Hz 音频，输入为堆叠的对数梅尔特征（80 个梅尔滤波器组，堆叠 2 帧），并在层 8 [2] 使用自条件化。CTC 词汇表由 384 个基于字符的单元构成。我们从 CTC 编码器的四个中间层（第 4、8、12 和 16 层）提取隐藏层表示，并沿特征维度拼接，以提供多尺度声学表示。

为了映射声学表示，我们训练了一个 1 层 Q-Former 投影器 [36]，将拼接后的隐藏层下采样 5 倍（从 15 帧窗口变为 3 个查询）。每个查询由其对应的 5 帧片段的均值池化表示构成，并通过可学习的位置嵌入对整个 15 帧窗口进行交叉注意力计算。Q-Former 的隐藏维度为 1024，前馈维度为 2048，包含 16 个注意力头，每个头的维度为 64。我们的基础语言模型是轻量级的 Granite 4.0 1B Base[37]，，我们使用秩为 128 的 LoRA 对其进行适配，并将 LoRA 应用于注意力层和多层感知机层。可训练参数（投影器 + 和 LoRA）的总数仅为 1400 万。对于 CTC 空白符号 ϵ，我们复用大语言模型词汇表中的 EOS 词元，以避免修改词表（任何特殊词元都可以）。

训练流程。 模型训练 3 个 epoch（总计 180K 步），使用 AdamW 优化器，峰值学习率为 3e-5，并采用带 5% 预热的余弦调度，最低学习率设为峰值学习率的 1%。我们采用 [2] 的平衡采样策略，其中 $\alpha \in [0, 1]$ 超参数控制数据集平衡程度，并设置为 $\alpha = 0.65$

（相较于 [2] 中的 $\alpha = 0.6$ ）适用于所有实验。复制正则化权重设为 $\lambda = 0.02$ （这是一个较小的值，可避免其主导 CTC 损失）。我们使用约 320 个话语的全局批次大小，将其分配到 8 个 H100 GPU 上。在 8 个 H100 GPU 上，训练 3 个 epoch 约需 30 小时；相比之下，自回归基线需 24 小时（由于交错填充增加了序列长度，耗时增加 1.25 倍）。

4.2. 数据集

我们的训练数据集包括 AMI（IHM+SDM） [38], Vox-Populi [39], YODAS（9K 小时子集） [40], CommonVoice15 [41], MLS [42], 财报22 [43], Fisher [44], CallHome [45], 和 SwitchBoard [46]。我们使用五种语言进行训练：英语、西班牙语、法语、德语和葡萄牙语。合并后的训练语料包含五种语言约 70K 小时的语音。评估时，我们使用训练数据集的官方验证集和测试集，以及另外三个仅用于测试的数据集：GigaSpeech [47], TED-LIUM [48], 和 SPGISpeech [49]。

训练期间，我们会过滤掉时长超过 80 秒的话语，以及字符帧比超过 0.66 的样本。我们使用 SpecAugment（时间和频率掩蔽），并以 25% 的概率，采用 MUSAN 和 FreeSound 背景音，在信噪比介于 -5 至 20 dB 的条件下进行噪声增强。

4.3. 评估方案

我们从转录准确率和推理效率两方面评估 NLE，重点关注 WER-RTFx 权衡。所有 RTFx 测量均通过单张 H100 图形处理器以 bf16 精度进行离线批量推理获得。对于自回归和非自回归模型，我们使用批量大小为 96 的话语。为评估模型在延迟敏感的对话场景中的表现，我们还测量了批量大小为 1 时的 RTFx；此时，自回归模型无法并行计算的问题会更加严重。对于 Open ASR 排行榜上的其他模型，我们使用排行榜代码库中经优化的开源实现，并在相同的 H100 硬件上进行测量，以确保公平比较。对于基于 CTC 的模型，我们采用 CTC 贪心解码（最大值索引+合并），因为束搜索带来的提升微乎其微，却会显著增加推理成本（我们观察到，WER 仅略有改善，而 RTFx 慢了 5–10 倍）。对于自回归模型，我们采用自回归贪心解码。WER 以小写形式计算，并应用 Whisper 规范化，使用 jiwer 库进行计算。我们报告了若干汇总结果：Open ASR 排行榜 [8]表示排行榜所含数据集的平均 WER；CV 和 MLS 指标分别为所有已报告的 CommonVoice 15 和多语种 LibriSpeech 子集的平均值。

5. 结果

5.1. OpenASR 排行榜比较

我们将 NLE 与提交时 Open ASR 排行榜上的前 6 个模型进行比较：Canary-Qwen 2.5B [50],Granite Speech 20亿和 80亿 [2], Phi-4 多模态 [51], Qwen3-ASR 1.7B [52], 以及 Parakeet 0.6B [50]。图 1 显示，NLE 位于帕累托前沿，平均 WER 排名第 4（5.79%），同时实时因子达到 1722。在前 6 个模型中，只有 Parakeet（实时因子为 4264）速度更快，但其 WER 更高（6.05%）。NLE 展现出更优的准确率—

表1： 词错误率（%，越低越好）和实时因子（越高越好）对比。*NLE* 的准确率可与自回归模型媲美，同时在批量/单句推理中分别快 *4×*/*27×* 倍。

数据集	NLE	AR	CTC
AMI-IHM	8.3	8.6	9.4
AMI-SDM	21.4	23.8	24.4
CV15-DE	5.6	4.7	6.3
CV15-EN	7.3	7.1	9.5
CV15-ES	5.0	4.1	5.5
CV15-FR	8.2	7.2	10.8
CV15-PT	3.0	2.7	3.4
财报	10.0	10.1	11.5
GIGASPEECH	10.1	10.0	10.6
LS-CLEAN	1.4	1.5	1.7
LS-OTHER	3.1	3.1	3.7
MLS-DE	4.7	4.5	4.9
MLS-EN	4.8	4.7	5.7
MLS-ES	3.5	3.1	3.7
MLS-FR	4.6	4.5	5.6
MLS-PT	10.0	10.1	8.5
SPGI	3.5	3.5	4.5
TED-LIUM	3.9	3.7	3.9
VOX	6.2	6.2	7.1
总体指标			
平均值（全部 19 项）	6.54	6.48	7.40
开放式自动语音识别平均值	5.79	5.82	6.55
Common Voice 平均值	5.79	5.18	7.10
MLS 平均值	5.51	5.39	5.66
速度指标			
实时因子：批量大小 96	1722	430	2584
实时因子：批量大小 1	322	12	760

速度权衡，相较于其他基于大语言模型的系统，NLE 在保持竞争力的准确率的同时，推理速度比 Canary-Qwen、Granite Speech、Qwen3-ASR 和 Phi-4 多模态等模型快 2 至 10 倍。值得注意的是，NLE 是 Open ASR 排行榜帕累托前沿上唯一支持多种语言的模型，支持英语、西班牙语、法语、德语和葡萄牙语五种语言；而 Canary-Qwen 和 Parakeet 仅支持英语。

5.1.1. NLE++（增强训练）

我们额外训练了**NLE++**，这是一个架构与数据相同、但包含多项训练改进的变体：更大的投影器（2层， 2×隐藏层与前馈维度），输入/输出丢弃为0.1，LoRA秩从128提高到160，2× 更高的峰值学习率（6e-5），5个训练轮次（对比3个），2×更大的全局批次大小（在2个节点的16块H100 GPU上共640条话语），以及最大音频长度延长至120秒（对比80秒）。更大的投影器和LoRA秩使可训练参数总数扩展至280M（160M投影器 + 120M LoRA），而NLE为14M。由于NLE++在训练预算和配置上不同于第5节中的受控设置，我们在表2中单独报告。NLE++ 取得了5.67%的开放式自动语音识别词错误率——较NLE绝对改进0.12%，并将全部19项平均值从6.54%改善至6.44%。由于投影器更大，实时因子小幅下降至1630（对比1722），但NLE++ 仍位于帕累托前沿（图1）。这些结果表明

表 2： *NLE* 与 *NLE++* 的总体词错误率（%）和实时因子对比。

	NLE	NLE++
总体指标		
平均值（全部 19 项）	6.54	6.44
开放式自动语音识别平均值	5.79	5.67
Common Voice 平均值	5.79	5.41
MLS 平均值	5.51	5.76
速度		
实时因子：批量大小 96	1722	1630
实时因子：批量大小 1	322	310

NLE能从扩大训练计算量和模型容量中获益，表明在更多资源下还有进一步改进的空间。

5.2. 受控评估

我们使用相同的训练设置，与两个基线进行受控比较：

• **仅使用 CTC 编码器：**从预训练语音编码器进行贪心解码。

• **受控自回归基线：**为在自回归与非自回归方法之间进行公平且受控的比较，我们使用上述相同的编码器、投影器、大语言模型骨干网络、数据集、平衡采样策略和优化设置，训练了一个基于大语言模型的自回归自动语音识别系统。两种系统的差异仅在于解码策略：自回归下一词元预测与非自回归编辑。自回归基线使用相同的投影器对声学嵌入进行投影，然后将其与从大语言模型词汇表中取出的、由序列起始标记和序列结束标记包裹的真实转录文本词元进行拼接。模型采用因果注意力模式，通过交叉熵损失进行标准的下一词元预测训练。在推理时，模型使用贪心解码以自回归方式生成转录文本。

表1展示了我们在覆盖5种语言的19个测试数据集上进行受控评估的结果。NLE始终优于CTC编码器基线，在保持较高推理速度的同时，将平均词错误率从7.40%降至6.54%。NLE持续改进了CTC的表现，在19个测试集中的17个上优于CTC。与受控自回归基线相比，NLE取得了相当的准确率，同时推理速度显著更快：在批处理场景下加速4倍（实时因子为1722，对比430），在单条语音设置下加速27倍（实时因子为322，对比12）。在无法进行批处理的实时对话应用中，延迟优势尤为明显，因为自回归模型无法并行化，这会成为关键瓶颈。需要指出的是，在CommonVoice子集上，NLE的表现不如自回归基线；在多语言场景下，其性能略差，这可能是因为CTC编码器的训练数据以英语为主，导致其生成的非英语识别假设较弱，而NLE随后必须对这些假设进行纠正，

再加上大语言模型分词器以英语为中心的 BPE 词汇表。这些结果证实了我们所提方法的有效性。

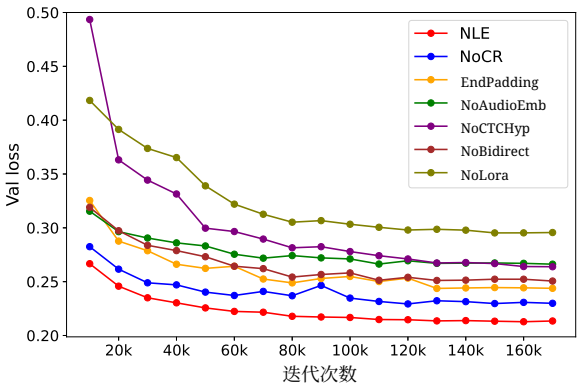


图3： 消融研究的训练步数与验证损失曲线（见第*5.3*节）。*NLE*（完整模型）的验证损失最低，证实每项设计选择都有助于提升整体性能。

5.3. 消融研究

图 3 分析了关键设计选择对验证损失的影响，而验证损失始终与测试集上更低的词错误率相关。

复制正则化（CR）损失。 移除CR损失（NoCR）会使验证损失变差，尽管报告的损失包含非负的CR项，这意味着即使增加了惩罚项，完整模型仍能达到更低的损失。这表明，CR通过明确鼓励恒等映射偏置，既提高了训练稳定性，也提升了最终性能。

双向注意力。 将大语言模型限制为因果注意力（NoBidirect），会限制其利用未来上下文进行编辑的能力。值得注意的是，尽管大语言模型是通过因果注意力预训练的，启用双向注意力仍能持续改善收敛效果和验证损失，这证实了未来上下文对于有效的非自回归编辑至关重要。

填充策略。 交错填充的表现显著优于序列末尾填充（EndPadding）。序列末尾填充会在插入时导致较大的词元位移，而交错填充则能保持词元局部性，与 Transformer 的局部性偏置相符。

音频条件化。 移除声学嵌入（NoAudioEmb；将其乘以 0）会大幅降低性能，这凸显了声学基础对于准确修正假设的必要性。

假设条件化。 移除作为条件的输入假设（NoCTCHyp；将其替换为等长的空白序列）也会降低性能，这表明模型难以从头预测整个假设。

LoRA 适配。 冻结大语言模型（NoLoRA）会使验证损失变差，这凸显了 LoRA 适配对于达到最佳性能的重要性。

5.4. 空白密度

我们研究了改变交错序列中插入槽位密度的影响。我们没有在每个词元之间插入一个空白（每 1 个），而是尝试每 2/3 个词元插入空白。表 3 表明，降低空白密度会降低准确率，同时仅带来极小的加速，因为序列长度主要由声学词元而非文本词元主导。在每个词元之间插入空白（每

表 3：空白密度（每 *K* 个词元添加一个插入槽）对平均词错误率和实时因子的影响。在每个词元之间插入空白（每 *1* 个）相比更稀疏的插入策略，能取得更优的词错误率，且实时因子下降可忽略不计。

空白密度	平均词错误率（%）	RTFx
每 1 个（NLE）	6.54	1722
每 2 个	6.80	1750
每 3 个	6.91	1770

表 4：多步编辑的平均词错误率与实时因子。第 *0* 步对应 *CTC* 基线（不进行编辑）。两步编辑带来的准确率提升可忽略不计，却会降低实时因子；而三步编辑会使性能降至低于单步结果。

编辑步数	平均词错误率（%）	RTFx
0（仅 CTC）	7.40	2584
1	6.54	1722
2	6.53	1259
3	6.59	1082

1 个）提供了最佳的准确率-速度权衡。

5.5. 多步编辑

我们研究迭代应用编辑器是否能进一步提高转录准确率。在多步编辑中，编辑器的输出对数几率先用 CTC（最大值索引 +折叠）解码，再与插入槽位重新交错，然后作为输入反馈给编辑器进行下一轮细化，同时保持声学嵌入固定。该过程可重复多次，每一步都可能纠正先前迭代中引入或遗漏的错误。表 4 表明，进行第二步编辑可带来适度改进，但会降低推理速度。然而，进一步迭代的收益递减，第三步还会使性能略有下降。这种下降可归因于分布不匹配：编辑器是在 CTC 假设上训练的，但在推理时会迭代处理自身输出，而这些输出具有不同的错误特征。训练期间的文本增强策略有望缓解这种不匹配，并改进多步细化。对大多数应用而言，单步编辑提供了最佳的准确率-速度权衡。

5.6. 误差分析

我们分析不同模型产生的错误类型，以了解其行为。图 4 展示了 CTC、NLE 和自回归模型的词错误率在插入、删除和替换错误中的构成，分别给出了所有数据集的平均值以及两个测试案例：AMI-SDM 和 MLS-PT。平均而言，自回归模型的插入率最高，这表明当声学证据较弱时可能会产生幻觉。具有挑战性的 AMI-SDM 数据集更鲜明地体现了这一模式：自回归模型产生的插入明显更多。相比之下，相比之下，NLE 的删除率最高、插入率最低，反映出其采用了更保守的编辑策略，倾向于删除而非插入。

有趣的是，在 MLS-PT 上，NLE 和自回归模型都未能优于 CTC，而 NLE 的替换错误尤为

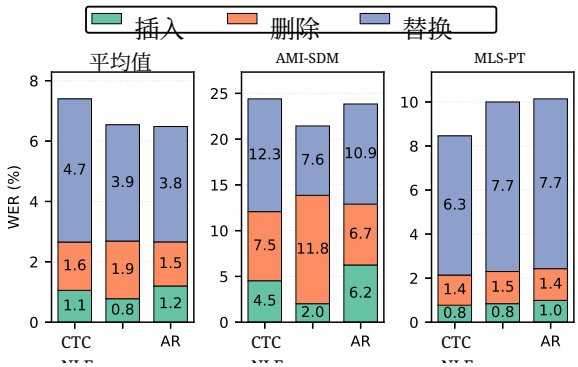


图 4：三种条件下的插入率、删除率和替换率（%）：所有数据集的平均值、*AMI-SDM* 和 *MLS-PT*。

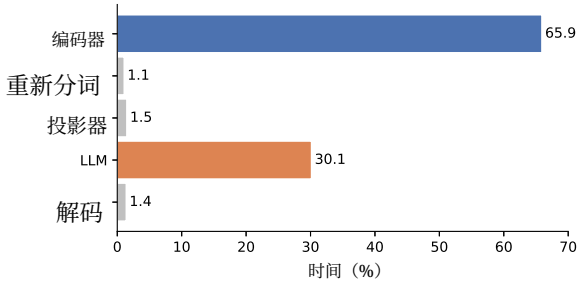


图5： *NLE* 各阶段的推理时间分解。编码器占总时间的 *66%*，大语言模型占 *∼30%*，其余所有阶段均低于 *4%*。

严重。MLS-PT 的训练集非常小，这使该数据集尤其具有挑战性，并可能导致过拟合或对葡萄牙语的适应不足。这表明，当 CTC 假设的质量相对较高时，编辑方法可能会引入错误，而非纠正错误，这凸显了初始假设质量的重要性。

5.7. 定性分析

为展示 NLE 所执行的纠错类型，表5列出了具有代表性的示例，对比 CTC 输出与 NLE 的预测结果。这些示例展示了 NLE 如何纠正多种错误类型：若干单词书写错误（示例 1）、替换纠正（示例 2），以及根据严重损坏的输入补全合理内容（示例 3）。示例 4–5 展示了其多语言能力：NLE 成功纠正了德语和法语转录文本。值得注意的是，在法语示例中，NLE 删除了一段较长的前言（书名、章节和旁白者信息）。虽然这段内容在语音中确实出现，但根据真实标签，它不应被转录。这表明模型已根据训练数据中的惯例学会过滤元数据内容。这些示例凸显了模型能够利用声学嵌入和预训练大语言模型提供的语言先验，从而生成多种语言中更准确的转录文本。

5.8. 推理时间分解

图5展示了 NLE 各阶段推理时间的分布。编码器前向传播占总时间的 66%，

表5：展示 *CTC* 输出和 *NLE* 纠正结果的编辑示例。错误以 红色高亮显示，纠正内容以蓝色显示，参考文本中的删除内容以橙色显示。

示例 1 **参考文本：** 我们将要设计的这个东西 是 一个新的遥控器 **CTC：** thesting weare ona design ish 一个新的遥控器 **NLE：** 我们将要设计的这个东西 一个新的遥控器
示例 2 **参考文本：** 他们经常被禁止上学，被迫 每天工作十八到二十个小时 **CTC：** 他们经常被禁止上学，被迫 设定 工作十八 到 每天二十个小时 **NLE：** 他们经常被禁止上学，被迫 工作十八到 每天二十个小时
示例 3 **参考：** 两者都极其重要，但创新需要得到 额外推动 **CTC：** 我说 极其重要，但创新需要得到 额外推动 **NLE：** 这些 都 极其重要，但创新需要 获得额外推动力
示例 4 **REF:** 顺便一提，他在其他大陆已经被誉为 重要的潮流引领者 **CTC:** 顺便一提，他在其他大陆 已经 重要的 潮流引领者 受到赞誉 **NLE:** 顺便说一句，他在其他大陆上已被当作 重要趋势受到庆祝
示例 5 **参考：** 耳朵，当诺˙厄将动物聚集˙于 方舟之前，他对自己说 **CTC：** 第十六章 te ab c 小伯爵 d·朱尔·迈特 录制 为了 '把牛羊听见时不˙耳朵聚集了˙这些 动物在方舟前，他心想 **NLE：** 耳朵当不˙耳聚集了˙方舟前的动物 方舟，他心想

计算量约为大语言模型的 2× 倍。值得注意的是，与编码器输出（下采样后为 10Hz，而非 50Hz）相比，大语言模型处理的序列更短 5×，这解释了这一结果。其余阶段——重新分词、投影和输出解码——合计仅占推理时间的 4%。相比之下，自回归系统中大语言模型会因顺序生成令牌而占据主导地位，而 NLE 的并行解码则使大语言模型的占比保持较低。

6. 讨论

我们提出了 NLE，一种基于大语言模型的非自回归自动语音识别系统，它将语音识别重构为条件转录编辑。NLE 利用预训练 CTC 编码器同时生成声学嵌入和初始假设，然后应用带有交错插入槽表示的双向大语言模型编辑器。NLE 的准确率与自回归基线相当，同时在批处理场景中实现高达 4 倍的推理加速，并且在对单语句延迟要求严格的对话场景中，加速优势进一步增大。我们的方法表明，预训练大语言模型可以通过轻量级 LoRA 适配器和注意力掩码修改，被有效地适配用于非自回归编辑。交错插入槽方案能够高效处理插入，同时保留 Transformer 架构的恒等映射偏置。

6.1. 局限性

与自回归模型相比，NLE 在处理输出与输入假设存在显著差异的任务时灵活性较低。NLE 擅长纠正局部错误，但不太可能泛化到需要大幅修改假设的任务，例如口语问答，因为预期回答与转录文本大相径庭。此外，当 CTC 编码器和大语言模型使用不同的分词器时，NLE 需要将假设从图形处理器传输到中央处理器以进行重新分词，然后再传回，这会略微增加延迟。虽然使用大语言模型的分词器进行 CTC 训练可以避免这一开销，但会显著增加编码器训练成本。

6.2. 未来工作

若干有前景的方向有望进一步提升 NLE 的能力。在训练期间采用文本增强策略，或可解决多步编辑中观察到的分布不匹配问题（第 5.5 节），从而有望实现更有效的迭代精炼。这类增强可以包括合成错误、改述，或将模型自身的预测用作训练输入，以更好地匹配推理时的条件。将编辑方法与掩码预测策略相结合，或可进一步提高转录准确率：模型可以迭代精炼不确定的预测，同时保持并行推理。这种混合方法可以兼具两种范式的优势：对大多数词元进行单遍编辑的效率，以及对难处理区域进行掩码精炼的能力。

另一个值得探索的方向是重构大语言模型架构，使其在不同层中分别处理音频和文本，并在不同模态之间使用交叉注意力，从而将计算复杂度相对于音频长度的平方级降至线性级。这对于长音频处理尤其有益。

未来研究的其他方向包括：

- 研究将底层大语言模型用作 CTC 波束解码的语言模型，以便在初始假设生成期间提供更强的语言先验。
- 对编码器和编辑器进行联合微调，以实现端到端优化（不过，这需要谨慎设计训练策略，以维持声学建模性能）。
- 通过能够处理部分假设和不完整声学上下文的机制，或采用基于分块的处理策略，在每个分块内保持并行推理优势，从而将该方法扩展到流式场景。

7. 参考文献

[1] C. Tang、W. Yu、G. Sun、X. Chen、T. Tan、W. Li、L. Lu、Z. MA 和 C. Zhang，《Salmonn：迈向具备通用听觉能力的大语言模型》，载于第十二届国际学习表征会议，2024。 [在线]。可访问：https://openreview.net/forum?id=14rn7HpKVk

[2] G. Saon、A. Dekel、A. Brooks、T. Nagano、A. Daniels、A. Satt、A. Mittal、B. Kingsbury、D. Haws、E. Morais 等，“Granite-

语音：具备强大英语自动语音识别能力的开源语音感知大语言模型，”*arXiv*预印本 *arXiv:2505.08699*，2025。

[3] A. Graves, S. Fern´andez, F. Gomez, 和J. Schmidhuber, 《连接主义时序分类：使用循环神经网络为未分段序列数据标注》，载于第*23*届国际机器学习会议论文集，2006年，第369–376页。

[4] S. Gong、S. Agarwal、Y. Zhang、J. Ye、L. Zheng、M. Li、C. An、P. Zhao、W. Bi、J. Han、H. Peng和L. Kong，“通过自回归模型适配扩展扩散语言模型”，载于第十三届国际学习表征会议，2025年。 [在线]。可获取：https://openreview.net/forum?id=j1tSLYKwg8[5] P. BehnamGhader、V. Adlakha、M. Mosbach、D. Bahdanau、N. Chapados和S. Reddy，“Llm2vec：大语言模型实为强大的文本编码器”，载于第一届语言建模会议，2024年。[6] C. Saharia、W. Chan、S. Saxena和M. Norouzi，“基于潜在对齐的非自回归机器翻译”，载于*2020*年自然语言处理实证方法会议论文集（*EMNLP*），2020年，第1098–1108页。[7] E. J. Hu、Y. Shen、P. Wallis、Z. Allen-Zhu、Y. Li、S. Wang、L. Wang、W. Chen等，“Lora：大语言模型的低秩自适应。”*ICLR*，第1卷，第2期，第3页，2022年。[8] V. Srivastav、S. Zheng、E. Bezzam、E. L. Bihan、A. Moumen和S. Gandhi，“Open ASR 排行榜：迈向可复现且透明的多语言语音识别评测”，*arXiv*预印本 *arXiv:2510.06961*，2025年。[9] Z. Ma、G. Yang、Y. Yang、Z. Gao、J. Wang、Z. Du、F. Yu、Q. Chen、S. Zheng、S. Zhang等，“一种简单得出人意料、具备强大自动语音识别能力的大语言模型方法”，*arXiv*预印本 *arXiv:2402 08846*，2024.。[10] Y. Chu、J. Xu、Q. Yang、H. Wei、X. Wei、Z. Guo、Y. Leng、Y. Lv、J. He、J. Lin等，“Qwen2-Audio技术报告”，*arXiv*预印本 *arXiv:2407.10759*，2024年。[11] A. Radford、J. W. Kim、T. Xu、G. Brockman、C. McLeavey和I. Sutskever，“通过大规模弱监督实现稳健语音识别”，载于国际机器学习会议。PMLR，2023年，第28 492–28 518页。[12] A. Graves，“基于循环神经网络的序列转导”，*arXiv*预印本 *arXiv:1211.3711*，2012年。[13] H. Xu、F. Jia、S. Majumdar、H. Huang、S. Watanabe和B. Ginsburg，“通过联合预测词元与时长实现高效序列转导”，载于国际机器学习会议。PMLR，2023年，第38 462–38 484页。[14] D. Rekesh、N. R. Koluguri、S. Kriman、S. Majumdar、V. Noroozi、H. Huang、O. Hrinchuk、K. Puvvada、A. Kumar、J. Balam等，“具有线性可扩展注意力的快速Conformer，用于高效语音识别”，载于*2023*年*IEEE*自动语音识别与理解研讨会（*ASRU*）。IEEE，2023年，第1–8页。[15]A. Hannun，“基于CTC的序列建模”，*Distill*，2017年，https://distill.pub/2017/ctc。[16] W. Chan、N. Jaitly、Q. V. Le和O. Vinyals，“Listen, Attend and Spell”，*arXiv*预印本 *arXiv:1508.01211*，2015年。[17] Y. Leng、X. Tan、W. Liu、K. Song、R. Wang、X.-Y. Li、T. Qin、E. Lin和T.-Y. Liu，“SoftCorrect：面向自动语音识别的软检测错误纠正”，载于*AAAI*人工智能会议论文集，第37卷，第11期，2023年，第13 034–13 042页。[18] Y. Higuchi、S. Watanabe、N. Chen、T. Ogawa和T. Kobayashi，“Mask CTC：结合CTC与掩码预测的非自回归端到端自动语音识别”，载于*Interspeech 2020*会议录，2020年，第3655–3659页。[19]Z. Fang、R. Zhang、Z. He、H. Wu和Y. Cao，“结合音系训练的非自回归中文自动语音识别错误纠正”，载于*2022*年计算语言学协会北美分会会议论文集：人类语言技术，2022年，第5907–5917页。

[20] M. Ghazvininejad、O. Levy、Y. Liu和L. Zettlemoyer，“掩码预测：条件掩码语言模型的并行解码”，载于*2019*年自然语言处理实证方法会议暨第九届自然语言处理国际联合会议论文集（*EMNLP-IJCNLP*），2019年，第6112–6121页。[21] W. Chan、C. Saharia、G. Hinton、M. Norouzi和N. Jaitly，“Imputer：通过插补与动态规划进行序列建模”，载于国际机器学习会议。PMLR，2020年，第1403–1413页。[22] Y. Leng、X. Tan、L. Zhu、J. Xu、R. Luo、L. Liu、T. Qin、X. Li、E. Lin和T.-Y. Liu，“Fastcorrect：面向自动语音识别的基于编辑对齐的快速错误纠正”，神经信息处理系统进展，第34卷，21 719，2021年，第21 708页–。[23] E. A. Chi、J. Salazar和K. Kirchhoff，“Align-refine：通过迭代式重新对齐实现非自回归语音识别”，载于*2021*年计算语言学协会北美分会会议论文集：人类语言技术，2021年，第1920–1927页。[24] L. Mangu、E. Brill和A. Stolcke，“在语音识别中寻求共识：词错误最小化及混淆网络的其他应用”，《计算机语音与语言》，第14卷，第4期，第373–400页，2000年。[25] H. Sak、M. Saraclar和T. G¨ung¨or，“面向实时自动语音识别的即时格重评分。”载于*Interspeech*，2010年，第2450–2453页。[26] K. Hu、T. N. Sainath、R. Pang和R. Prabhavalkar，“基于审议模型的双遍端到端语音识别”，载于*ICASSP 2020—2020*年*IEEE*声学、语音与信号处理国际会议（*ICASSP*）。IEEE，2020年，第7799–7803页。[27] R. Ma、M. Qian、P. Manakul、M. Gales和K. Knill，“生成式大语言模型能否进行自动语音识别错误纠正？”*arXiv*预印本 *arXiv:2307.04172*，2023年。

[28] R. Ma, M. Qian, M. Gales, and K. Knill, 《自动语音识别错误纠正 using 大语言模型，” *IEEE* 音频、语音与语言处理汇刊，2025。

[29] R. Ma, M. J. Gales, K. M. Knill, 和M. Qian, “N-最佳T5：Ro-利用多个输入识别假设和受限解码空间提升自动语音识别错误纠正，” *arXiv* 预印本 *arXiv:2303.00456*，2023。

[30] J. Gu、J. Bradbury、C. Xiong、V. O. Li 和 R. Socher，《非自回归神经机器翻译》，载于国际学习表征会议，2018 年。[31] J. Gu、C. Wang 和 J. Zhao，《莱文斯坦变换器》，载于神经信息处理系统进展，第 32 卷，2019 年。[32] Y. Zhang、Y. Zhang、L. Cui 和 G. Fu，《采用感知复制的潜在对齐的非自回归文本编辑》，载于*2023* 年自然语言处理实证方法会议论文集，2023 年，第 7075–7085 页。[33] K. He、X. Zhang、S. Ren 和 J. Sun，《用于图像识别的深度残差学习》，载于*IEEE*计算机视觉与模式识别会议论文集，2016 年，第 770–778 页。[34] ——，《深度残差网络中的恒等映射》，载于欧洲计算机视觉会议。施普林格，2016 年，第 630–645 页。[35] A. Gulati、J. Qin、C.-C. Chiu、N. Parmar、Y. Zhang、J. Yu、W. Han、S. Wang、Z. Zhang、Y. Wu等，《Conformer：用于语音识别的卷积增强型 Transformer》，载于*Interspeech 2020* 论文集，2020 年，第 5036–5040 页。[36] J. Li、D. Li、S. Savarese 和 S. Hoi，《BLIP-2：利用冻结图像编码器和大语言模型自举语言-图像预训练》，载于国际机器学习会议。PMLR，2023 年，第 19 730–19 742 页。[37] IBM Research，《Granite 4.0语言模型》，https://github.com/ibm-granite/granite-4.0-language-models，2025 年，访问日期：2025-10-01。

[38] W. Kraaij、T. Hain、M. Lincoln 和 W. Post，《AMI会议语料库》，载于行为研究方法与技术国际会议论文集，2005 年，第 1–4 页。[39] C. Wang、M. Riviere、A. Lee、A. Wu、C. Talnikar、D. Haziza、M. Williamson、J. Pino 和 E. Dupoux，《VoxPopuli：用于表征学习、半监督学习和解释的大规模多语言语音语料库》，载于计算语言学协会第 *59* 届年会暨自然语言处理第 *11* 届国际联合会议论文集（第 *1* 卷：长篇论文），2021 年，第 993–1003 页。[40] X. Li、S. Takamichi、T. Saeki、W. Chen、S. Shiota 和 S. Watanabe，《YODAS：面向YouTube的音频与语音数据集》，载于*2023* 年 *IEEE* 自动语音识别与理解研讨会（*ASRU*）。IEEE，2023 年，第 1–8 页。[41] R. Ardila、M. Branson、K. Davis、M. Kohler、J. Meyer、M. Henretty、R. Morais、L. Saunders、F. Tyers 和 G. Weber，《Common Voice：大规模多语言语音语料库》，载于第十二届语言资源与评测会议论文集，2020 年，第 4218–4222 页。[42] V. Pratap、Q. Xu、A. Sriram、G. Synnaeve 和 R. Collobert，《MLS：用于语音研究的大规模多语言数据集》，载于*Interspeech 2020* 论文集，2020 年，第 2757–2761 页。[43] M. Del Rio、P. Ha、Q. McNamara、C. Miller 和 S. Chandra，《Earnings-22：真实环境中口音的实用基准》，*arXiv* 预印本 *arXiv:2203.15591*，2022 年。[44] C. Cieri、D. Miller 和 K. Walker，《Fisher语料库：面向下一代语音转文本技术的资源》，载于第四届语言资源与评测国际会议（*LREC*）论文集，2004 年，第 69–71 页。[45] A. Canavan 、 D. Graff 和 G. Zipperlen，《CallHome 美式英语语音》，网络下载，费城，1997 年，l DC97S42。[46] J. J. Godfrey、E. C. Holliman 和 J. McDaniel，《Switchboard：用于研究与开发的电话语音语料库》，载于 *[*论文集 *] ICASSP-92*：*1992* 年 *IEEE* 声学、语音与信号处理国际会议，第 1 卷。IEEE，1992 年，第 517–520 页。[47] G. Chen、S. Chai、G.-B. Wang、J. Du、W.-Q. Zhang、C. Weng、D. Su、D. Povey、J. Trmal、J. Zhang等，《GigaSpeech：不断扩展的多领域自动语音识别语料库，包含 10,000 小时转录音频》，载于*Interspeech 2021* 论文集，2021 年，第 3670–3674 页。[48] A. Rousseau、P. Del´eglise 和 Y. Esteve，《TED-LIUM：专用于自动语音识别的语料库》，载于*LREC*，2012 年，第 125–129 页。[49] P. K. O’Neill、V. Lavrukhin、S. Majumdar、V. Noroozi、Y. Zhang、O. Kuchaiev、J. Balam、Y. Dovzhenko、K. Freyberg、M. D. Shulman等，《SPGISpeech：5,000 小时转录财报音频，用于全格式端到端语音识别》，载于*Interspeech 2021* 论文集，2021 年，第 1434–1438 页。[50] M. Sekoyan、N. R. Koluguri、N. Tadevosyan、P. Zelasko、T. Bartley、N. Karpov、J. Balam 和 B. Ginsburg，《Canary-1b-v2 与 parakeet-tdt-0.6 b-v3：高效高性能的多语言自动语音识别和自动语音翻译模型》，*arXiv* 预印本 *arXiv:2509.14128*，2025 年。[51] A. Abouelenin、A. Ashfaq、A. Atkinson、H. Awadalla、N. Bach、J. Bao、A. Benhaim、M. Cai、V. Chaudhary、C. Chen等，《Phi-4-mini 技术报告：通过 LoRA 混合实现紧凑而强大的多模态语言模型》，*arXiv* 预印本 *arXiv:2503.01743*，2025 年。[52] X. Shi、X. Wang、Z. Guo、Y. Wang、P. Zhang、X. Zhang、Z. Guo、H. Hao、Y. Xi、B. Yang等，《Qwen3-ASR 技术报告》，*arXiv* 预印本 *arXiv:2601.21337*，2026 年。