CASS-NAT: CTC ALIGNMENT-BASED SINGLE STEP NON-AUTOREGRESSIVE TRANSFORMER FOR SPEECH RECOGNITION 复现报告

何旭杰

2024.12.30

摘要

随着非自回归模型（Non-Autoregressive Models, NAT）在语音识别领域的研究逐渐深入，其高效解码特性成为关注的焦点。借助 Kaldi 工具，以及 ESPnet 和 Harvard NLP 团队开源的模块化、插件化的模型训练框架，结合 Conformer 模型与改进的数据增强策略，本文复现并优化了文献中的自回归模型（AT）和单步解码的非自回归模型（CASS-NAT）。实验结果显示，复现的 CASS-NAT 模型在 LibriSpeech 和 Aishell1 数据集上均表现优异，特别是在中文语料库 Aishell1 上，开发集和测试集的 CER 分别比文献结果降低了 0.4% 和 0.5%。此外，复现的 AT 模型性能接近文献基准，进一步验证了复现工作的可靠性。通过实验分析，本文证明了引入 Conformer 模型和增强速度扰动策略的有效性。本研究不仅验证了 CASS-NAT 方法的可行性与鲁棒性，还通过优化提升了其性能，为高效语音识别的研究提供了新思路，并为非自回归模型在多语言场景中的应用奠定了基础。

关键词： End-to-end speech recognition；NAT；Single step generation；CTC alignments

1 引言

近年来，随着深度学习的快速发展，自动语音识别（Automatic Speech Recognition, ASR）技术取得了显著的进步，其广泛应用于智能助手、实时翻译和语音搜索等领域。然而，传统的自回归模型（Autoregressive Models）在解码过程中具有较高的时间复杂度，导致在实时性和效率要求较高的场景中表现受限。因此，非自回归模型（Non-Autoregressive Models, NAT）逐渐成为语音识别领域的研究热点。

基于连接时序分类（Connectionist Temporal Classification, CTC）的非自回归语音识别模型 CASS-NAT，通过结合 CTC 对齐和非自回归解码，显著减少了解码步骤并提升了推理速度。同时，其架构引入了混合注意力机制和触发器掩码，为模型提供了更强的上下文捕获能力。相比于传统方法，CASS-NAT 在保证识别精度的同时大幅提升了解码效率。

本研究基于 CASS-NAT 的核心思想，探索其在实际语音识别任务中的表现。通过复现和分析相关实验，我们不仅能够验证其在标准数据集上的性能，还可以为进一步优化非自回归

语音识别模型提供参考。这一研究对于推动高效语音识别模型的应用具有重要意义，同时也为其他语言理解任务中的非自回归建模提供了借鉴。

2 相关工作

近年来，端到端（E2E）语音识别模型与混合系统相比已经显示出有竞争力的结果 [14,15]。这些模型包括基于注意的编码器-解码器（AED）[3,6,27]，连接时序分类（CTC）[8] 及其变体 [18,20,23]。

2.1 自回归模型

自回归模型（Autoregressive Models）是语音识别领域的核心技术之一，其基本思想是通过条件概率链式规则逐步生成输出序列。在这种方法中，每个时间步的预测都依赖于之前所有已生成的输出，常见的自回归架构包括基于序列到序列（Sequence-to-Sequence, Seq2Seq）的模型以及基于注意力机制（Attention Mechanism）的变体。

典型的自回归语音识别模型采用编码器-解码器（Encoder-Decoder）结构，编码器将输入的语音特征转化为高维表示，解码器通过逐步生成输出字符或单词完成语音到文本的转换。此类模型的代表包括基于循环神经网络（Recurrent Neural Networks, RNNs）的 Listen, Attend and Spell（LAS）模型，以及结合了自注意力机制的 Transformer 模型。自回归模型的生成方式使其能够很好地捕获上下文信息，从而保证输出序列的流畅性和准确性。

然而，自回归模型存在显著的效率瓶颈。由于生成过程是严格的序列化操作，解码时间随序列长度线性增加，这在实时应用中限制了其使用。同时，生成过程对前一步预测的强依赖性也使得模型容易受到误差累积（Error Propagation）的影响。因此，尽管自回归模型在性能上有较好的表现，其高时间复杂度和推理效率问题为非自回归模型的发展提供了动机。

2.2 非自回归模型

AED 中的解码器自回归生成输出序列。为了加速输出序列的并行生成，提出了非自回归变压器（NAT）[9,19,21]，根据并行生成的迭代次数，NAT 可以分为迭代 NAT 和单步 NAT。流行的迭代 NATs 将解码器视为一种掩码语言模型，首先对低置信度的 token 进行掩码，然后从未掩码的令牌中预测新的 token。

非自回归模型（Non-Autoregressive Models, NAT）是近年来语音识别领域的一项重要研究方向，旨在克服传统自回归模型在解码效率上的限制。与自回归模型逐步生成输出序列的方式不同，非自回归模型通过并行生成整个输出序列，大幅减少了解码时间。其核心思想是消除输出序列中各时间步之间的依赖性，从而实现高效推理。

非自回归模型通常采用连接时序分类（Connectionist Temporal Classification, CTC）或插值序列预测（Insertion-based Prediction）等方法。CTC 通过对齐机制，允许模型在输入与输出序列长度不同的情况下进行训练；而插值预测方法则通过插入生成策略逐步完善初始序列。此外，一些非自回归模型还结合了掩码生成策略和多路径解码技术，以进一步提高生成质量。

对于 NAT 模型来说，解码器输入的长度预测是一个难题。为了解决这个问题，Higuchi 等人利用 CTC [11] 的识别结果长度，而 Chan 等人提出了一种名为 Imputer 的模型，该模型

直接使用了输入特征序列的长度 [4]。Bai 等人使用固定长度的解码器输入来提取解码器输入的数据驱动声学表示 [1]。Tian 等人仅使用带有 CTC 尖峰的编码器输出作为解码器输入 [22]。

3 本文方法

3.1 本文方法概述

本文没有使用数据驱动的或没有物理意义的不完整的声学表示，而是试图提取 Token 级声学嵌入，为每个 Token 提供更完整的声学表示，以更有效地捕获语言语义。因此，本文提出了一种基于 CTC 对齐的单步非自回归 Transformer（CASS-NAT），旨在解决非自回归语音识别模型中由于输入与输出长度不匹配带来的问题。通过引入 CTC 对齐信息，本文方法能够并行地为每个输出令牌提取声学嵌入，并将其替代传统自回归模型中的词嵌入。该方法结合了 Transformer 的高效结构，同时通过引入误差驱动的对齐采样策略进一步降低了解码错误率。

本文的核心改进包括三个方面：使用 CTC 对齐信息生成触发掩码以精确定位声学范围；通过触发掩码和位置编码提取令牌级声学嵌入；提出误差驱动的对齐采样方法，兼顾了解码精度和并行性。此外，本文方法通过联合优化 CTC 损失和交叉熵损失的方式实现训练，有效提升了模型性能。

本文提出用于语音识别的基于 CTC 对齐的单步非自回归的 Transformer，其架构基于 CTC-Attention 混合体系结构和 Transformer 结构 [25]。

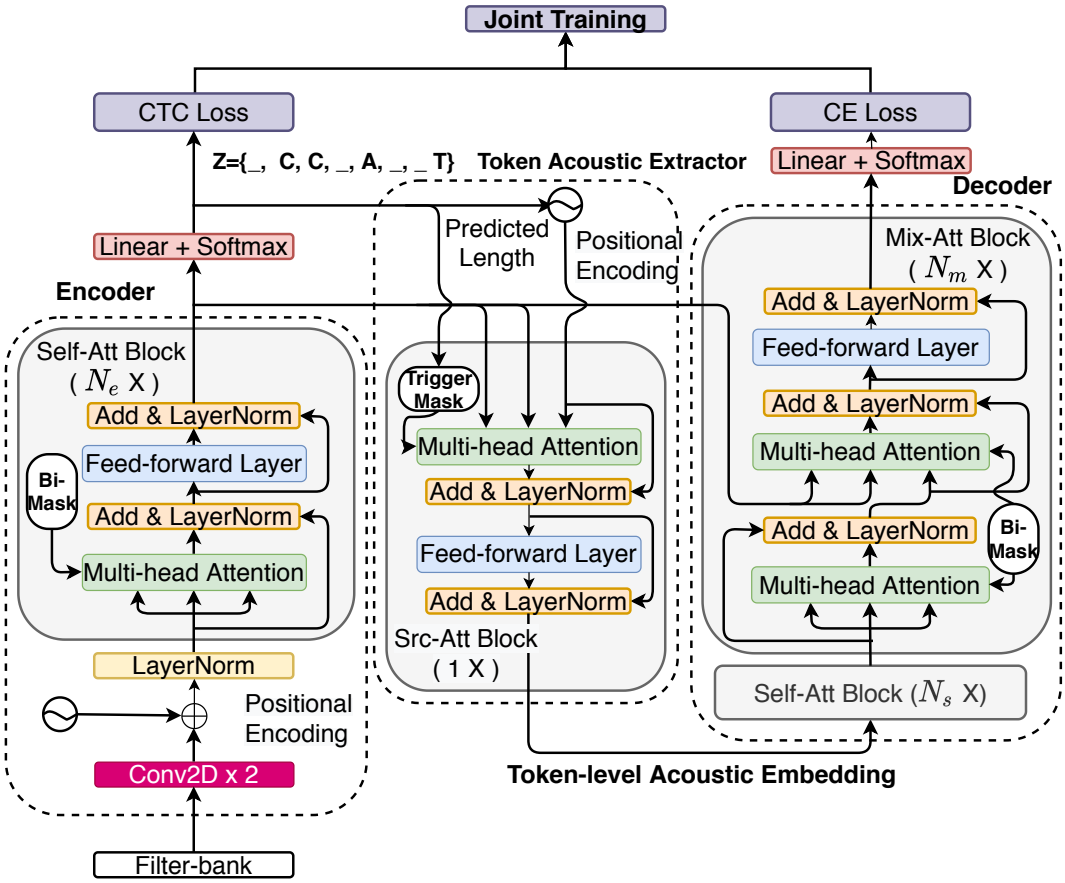

图 1. CASS-NAT 架构

单步非自回归 Transformer 只需经历一次迭代即生成输出序列，其中假设语义可以通过声学表示获得，对于每个输出使用声学表示来替代词嵌入。CTC 对齐包含了两种信息：(a) 解码器输入的 tokens 数量，(b) 每个 token 的声学时间跨度。这些信息会被用来并行提取每个 token 的声学表示，称为 token 层次的声学嵌入。为了使解码器并行生成，这些声学嵌入将会代替自回归 Transformer 中的词嵌入。推理阶段，CTC 对齐的输出空间使用基于错误率对齐的采样方法来降低错词率并保持并行性。CTC 对齐首先转化为触发掩码来标记每个 token 对应的音频时间段，帮助模型确定音频信号的时间范围和每个 token 的位置信息，通过结合触发掩码、位置编码以及 encoder-decoder 的注意力机制，模型可以同一时间并行提取每个令牌的声学嵌入。训练阶段，使用 Viterbi 对齐来作为目标函数的最大接近值。

3.2 特征提取模块

特征提取模块的目标是从原始语音信号中提取高层次的特征表示，以便后续的编码器和解码器模块能够最佳地利用这些信息。

3.2.1 语音信号预处理

输入信号首先被处理为 80 维的梅尔滤波特征（Mel-filter bank features），每 10ms 计算一次，并使用 25ms 的滑动窗。为了捕捉更长时间范围的声学模式，按照每 3 个连续帧进行拼接的方式，生成 240 维的特征向量。为了降低时间维度上的凸余信息，模块中加入了两层卷积（Conv2D），每层包含 64 个滤波器，卷积核大小为 3，步帧为 2。这种设计实现了 4 倍时间下采，同时增强了局部特征。

3.2.2 声学嵌入提取器

声学嵌入提取器利用 CTC 对齐信息辅助提取 token 级的声学嵌入。

- **触发掩码生成：** CTC 对齐的第一个信息为每个 token 对应的声学时间跨度，通过定义从 CTC 对齐到触发掩码来利用该信息，其中每个 token 的第一个索引即为其结束边界。例如，若 CTC 对齐结果为 $Z = \{\_, C, C, \_, A, \_, \_, T, \_\}$，其中 _ 为空白符号，则 Token A 对应生成的触发掩码为 $[0, 0, 1, 1, 1, 0, 0, 0, 0]$，该映射在训练和解码中是一样的。
- **声学嵌入提取：** CTC 对齐的第二个信息为解码器输入的 Tokens 数量。将空白符号 _ 和重复去除之后，Z 中的符号数量作为正弦位置编码的预测长度，也就是解码器输入的长度，最后通过一层源注意力机制提取 Token 级声学嵌入，从而保证嵌入表示的准确性和可并行性，另外，在训练中使用 viterbi-alignment 来解决长度不匹配的问题。

3.3 损失函数定义

在本文模型框架中，CTC 对齐 Z 作为潜在变量被引入。假设输入序列为 $X = \{x_1, ..., x_t, ..., x_T\}$，目标结果为 $Y = \{y_1, ..., y_u, ..., y_U\}$，然后将目标函数分解为：

$$\log P(Y|X) = \log E_{Z|X}[P(Y|Z, X)], \quad Z \in q \tag{1}$$

其中 q 是可以映射到 Y 的对齐集合，Y 是 CTC 输出空间中所有对齐的子集。为了减少计算成本，使用最大近似 [26]：

$$\log P(Y|X) \geq E_{Z|X}[\log P(Y|Z,X)] \tag{2}$$

$$\approx \max_{Z} \log \prod_{u=1}^{U} P(y_u|z_{t_{u-1}+1:t_u}, x_{1:T}) \tag{3}$$

其中 t_u 是 token u 的结束边界，$t_0 = 0$，对齐 Z 告诉解码器在哪里提取每个标记的声学表示作为辅助信息。与 [1] 中的假设相同，声学嵌入也可以用来捕获语言语义。然后，通过以任务比 λ [25] 共同最大化解码器损失和编码器侧的 CTC 损失来训练框架，联合损失函数 L_{joint}：

$$L_{\text{joint}} = L_{\text{dec}} + \lambda \cdot L_{\text{CTC}} \tag{4}$$

其中：

- L_{dec} 是解码器的交叉熵损失，通过最大化 token 级声学嵌入的后验概率实现：

$$L_{\text{dec}} = \max_{Z} \log \prod_{u=1}^{U} P(y_u|z_{t_{u-1}+1:t_u}, X), \tag{5}$$

  其中 Z 表示 CTC 对齐结果，t_u 是 token u 的结束时间边界。

- L_{enc} 是编码器损失，用于优化编码器的输出对齐。

$$L_{enc} = \log \sum_{Z \in q} \prod_{i=1}^{T} P(z_i|X) \tag{6}$$

联合损失的设计既保证了解码器的语言建模能力，又通过 CTC 对齐信息提升了模型的对齐精度。

3.4 推理中基于误差的采样解码

在解码过程中，获得接近假设的 viterbi-alignment 的 CTC 对齐是至关重要的。本文提出了一种基于误差的对准采样方法，与最佳路径对齐（best path alignment，BPA）和集束搜索对齐（beam searched alignment，BSA）相比，该方法在 WER 和 RTF 方面更有效，称为 ESA。为了生成 ESA，首先选择 Top1 概率较低的 CTC 输出（经验设置阈值为 0.7），如图2中蓝色部分所示。然后在每个选定的输出中随机选择 top-2 令牌，而其他输出继续使用 Top-1 令牌。与 BPA 相比，采样的对齐有可能更接近 oracle 对齐。与 BSA 相比，采样可以并行实现，避免了 RTF 的增加。ESA 与 BPA 的解码器输入长度不同。如图 2 所示，BPA 的长度为 4，ESA 的长度为 3、5、4、5。对齐中 Token 数的波动允许 ESA 方法对与 oracle 对齐长度相同的对齐进行采样。

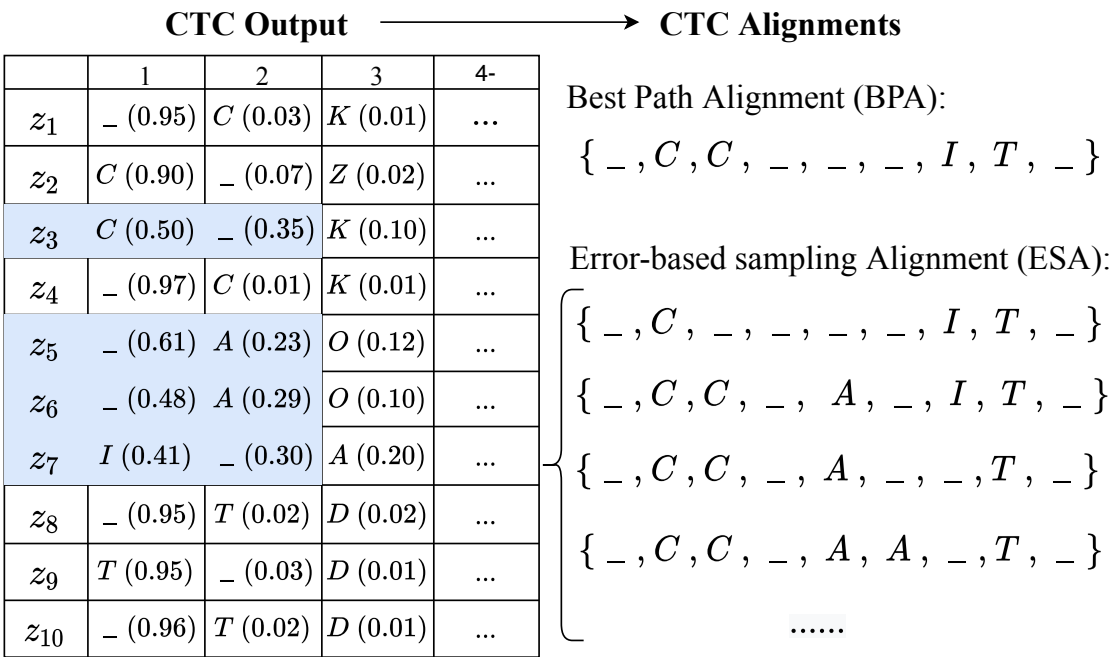

图 2. 基于错误率的对齐采样方法，$C(0.90)$ $P(z_i = C|X) = 0.90$

4 复现细节

4.1 与已有开源代码对比

本次复现工作的代码主要参考了两个开源项目，分别针对音频数据的预处理和核心模型的实现进行了借鉴与扩展。

首先，在音频数据的预处理部分，参考了 Kaldi 并基于其框架进行二次开发，以生成符合 Kaldi 标准的数据文件格式。Kaldi 提供了丰富且高效的音频处理工具，本次工作中主要参考的代码包括以下模块：

- **utt2spk**：从音频文件中提取音频 ID 和对应的说话人映射关系，用于后续的数据组织和标注。
- **spk2gender**：提取说话人的性别映射信息，支持基于性别的分析与处理。
- **perturb_data_dir_speed**：对音频数据进行速度扰动处理，生成多样化的语音样本，以增强模型对不同语速的鲁棒性。
- **make_fbank**：提取音频的梅尔滤波特征（FBANK），生成特征文件路径 `feats.scp` 和音频帧数文件 `utt2num_frames`，为后续模型训练提供特征输入。
- **compute-cmvn-stats**：计算全局特征均值和方差（CMVN），标准化输入特征以提高模型的稳定性和收敛性。

其次，在核心代码实现方面，借鉴了开源项目 ESPnet [24] 和 Harvard NLP 团队的 Transformer 实现。这些开源项目提供了高度模块化和插件化的工程架构，使得代码的扩展和维护更加便捷。通过参考这些成熟的代码框架，本次工作在其基础上进行了改进与二次开发，主要包括以下内容：

- 工程架构的借鉴：ESPnet 和 Harvard NLP 团队的代码均以插件化为核心，模块之间解耦性极高。各功能模块（如模型、任务、工具等）可以独立开发和扩展，新模型仅需添加对应的配置文件，即可通过现有脚本实现训练和推理。
- 添加 Conformer 模型：在原有框架的基础上，集成了 Conformer 模型。通过保留框架的插件化特点，用户只需简单配置相关参数，即可便捷地训练 Conformer 模型。
- 优化训练过程：结合 ESPnet 的多任务训练策略和 Harvard NLP 团队的 Transformer 实现，进一步优化了训练过程中的超参数设置与模型调优方法，从而提升模型性能和训练效率。

4.2 实验环境搭建

4.2.1 硬件环境

- CPU：双路 Intel Xeon Platinum 8358P，64 核 128 线程
- GPU：NVIDIA A100，40GB ×2
- 主板：Supermicro X12DPG-QT6
- 内存：512GB DDR4 3200MHz

4.2.2 软件环境

- Linux: Ubuntu 20.04 LTS
- Python 版本：3.7
- Pytorch 版本：1.2
- 重要辅助库：Librosa、Kaldi 等

4.2.3 数据集准备

实验基于 LibriSpeech v1.1（960 小时英文语音）和 Aishell1（178 小时中文语音）数据集。输入特征为 80 维梅尔滤波特征，处理步骤包括帧拼接（每 3 帧生成 1 个特征向量）和数据增强（SpecAugment 和速度扰动）。

4.3 创新点

4.3.1 使用 Conformer 替代 Transformer

在复现过程中，针对语音识别任务的特性，将原论文中的 Transformer 编码器模块替换为 Conformer [10]。相比于 Transformer，Conformer 更加适合语音信号的建模需求，其结合了卷积模块和多头自注意力机制，能够同时捕获局部和全局特征。Conformer 模块采用了 12 层堆叠结构，每层包含多头自注意力、卷积模块、前馈网络以及残差连接。卷积模块的核大小设置为 31，以增强局部特征提取能力。

4.3.2 添加额外速度扰动

语音信号在真实场景中经常受到语速变化的影响，而标准的数据增强方法可能不足以涵盖所有的语速变化情况，速度扰动可以模拟不同语速的语音数据，帮助模型更好地泛化到语速变化的场景，因此，为了进一步提升模型对语速变化的鲁棒性，在复现中通过使用 Kaldi 工具，增加了额外的速度扰动策略。在常规的速度扰动基础上，新增了更丰富的速度变化比例，使用了 $[0.9, 1.0, 1.1]$ 的默认速度扰动比例，并额外加入了 $[0.8, 1.2]$ 两个极端速度扰动比例，以进一步扩大语速变化范围。

5 实验结果分析

实验在 960 小时的 Librisspeech 英语语料库 [16] 和 178 小时的 Aishell-1 普通话语料库 [2] 上进行，实验不使用额外的语言模型。输出标签集由 Librisspeech 的 sentencepece 方法 [13] 获得的 5k 个单词和从训练集获得的 aishel-1 的 4230 个汉字组成。首先训练 CTC/Attention AT 基线，使用与 Libisspeech 的 [12] 相同的架构 ($N_e = 12, N_d = 6, d_{FF} = 2048, H = 8, d_{MHA} = 512$)。对于 CASS-NAT，将 AT 中的解码器替换为 $N_s = 3$ 个自注意块和 $N_m = 4$ 个混合注意块。对于 Aishell-1，AT 基线和 CASS-NAT 的架构与用于 Librispeech 的对应相同，除了 $N_e = 6$。对于所有方法，编码器中的两个 CNN 各有 64 个滤波器，核大小为 3，步长为 2。对于所有模型，采用 [17] 中的 Double 调度作为学习调度，其中学习率上升到并保持在 0.001，然后呈指数衰减到 1e5。归一化层中的退出率为 0.1，标签平滑的惩罚值为 0.1。对于所有模型，损失函数公式4中的任务比 λ 设置为 1。实验使用开发集来使训练能够提前停止，大多数实验在 50 个周期内结束，并使用模型平均进行最终评估。

表 1. 不同模型在 LibriSpeech 数据集上的性能比较 (WER %)

模型	**Type**	**WER (%)**			
		dev-clean	**dev-other**	**test-clean**	**test-other**
RETURNN [15]	AT	4.3	12.9	4.4	13.5
AT(Paper)	AT	3.4	8.5	3.6	8.5
AT(Reproduction)	AT	3.8	9.1	4.2	9.5
Imputer [4]	NAT	-	-	4.0	11.1
CASS-NAT(Paper)	NAT	3.7	9.2	3.8	9.1
CASS-NAT(Reproduction)	NAT	3.6	10.0	4.5	11.6

表1中对 AT 基线和 NAT 模型在英文数据集 Librisspeech 上进行实验的结果进行对比，与 AT 基线相比，CASS-NAT 的 WER 相对高 6%，复现结果在标准数据集（如 dev-clean 和 test-clean）上的表现接近文献结果，但在噪声更大的 dev-other 和 test-other 数据集上的性能略有下降。

在表 2 中对现有工作、文献的实验结果与复现在中文数据集 Aishell-1 上进行实验的结果进行对比。相比文献中的结果，改进后的实验结果有着更低的 WER，低于 AT 基线和其他 NAT 方法中报告的 WER，这表明所提出方法的有效性和可推广性。

表 2. 不同模型在 Aishell1 数据集上的性能比较 (CER %)

模型	NAT Type	Dev	Test
Masked-NAT [5]	iterative	6.4	7.1
Insertion-NAT [7]	iterative	6.1	6.7
ST-NAT [22]	single step	6.9	7.7
LASO [1]	single step	5.8	6.4
AT(Paper)	n/a	5.5	5.9
AT(Reproduction)	n/a	4.7	5.1
CASS-NAT(Paper)	single step	5.3	5.8
CASS-NAT(Reproduction)	single step	4.9	5.3

6 总结与展望

本文复现了文献中的两种主要模型——自回归模型（AT）和基于单步非自回归解码的 CASS-NAT，并在 LibriSpeech 和 Aishell1 数据集上进行了实验验证。复现结果表明，通过适当的优化与调整，复现的模型性能在大多数评估指标上均超过了文献中的结果，尤其是在 Aishell1 数据集上的表现显著优于文献中的基准。

实验结果分析表明，AT 模型在语音识别任务中依然表现出了较高的准确性，但其解码速度受限于自回归特性。相比之下，CASS-NAT 模型利用非自回归解码的优势，在解码速度上实现了显著提升，同时通过结合 CTC 对齐信息有效降低了解码过程中的错误率，从而进一步缩小了与自回归模型的性能差距。这一结果表明，CASS-NAT 是实现高效语音识别的一个有潜力的方向。

在未来的工作中，可以从以下几个方面进行改进和扩展：

- 模型架构的改进：进一步优化 CASS-NAT 的解码策略，例如引入动态调整的触发掩码机制，或结合更多上下文信息以提高模型的生成能力。
- 多语言场景的适配：当前实验仅针对英语和中文数据集，未来可以将 CASS-NAT 应用于更多语言或多语言场景，验证其在复杂语音识别任务中的适应性。
- 实际应用中的部署：结合推理速度优化，探索在低资源设备上的部署和性能表现，评估其在实际应用中的可行性。
- 结合预训练模型：尝试结合大型预训练语音模型（如 Wav2Vec 2.0 或 Whisper）以进一步提升 CASS-NAT 的特征建模能力。

参考文献

[1] Ye Bai, Jiangyan Yi, Jianhua Tao, Zhengkun Tian, Zhengqi Wen, and Shuai Zhang. Listen attentively, and spell once: Whole sentence generation via a non-autoregressive architecture for low-latency speech recognition. *Proc. Interspeech 2020*, pages 3381–3385, 2020.

[2] Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *20th O-COCOSDA*, pages 1–5. IEEE, 2017.

[3] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *ICASSP*, pages 4960–4964. IEEE, 2016.

[4] William Chan, Chitwan Saharia, Geoffrey Hinton, Mohammad Norouzi, and Navdeep Jaitly. Imputer: Sequence modelling via imputation and dynamic programming. In *International Conference on Machine Learning*, pages 1403–1413. PMLR, 2020.

[5] Nanxin Chen, Shinji Watanabe, Jesús Villalba, and Najim Dehak. Non-autoregressive transformer automatic speech recognition. *arXiv preprint arXiv:1911.04908*, 2019.

[6] Linhao Dong, Shuang Xu, and Bo Xu. Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In *ICASSP*, pages 5884–5888. IEEE, 2018.

[7] Yuya Fujita, Shinji Watanabe, Motoi Omachi, and Xuankai Chang. Insertion-based modeling for end-to-end automatic speech recognition. *Proc. Interspeech 2020*, pages 3660–3664, 2020.

[8] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376, 2006.

[9] Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. In *ICLR*, 2018.

[10] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *Proc. Interspeech 2020*, pages 5036–5040, 2020.

[11] Yosuke Higuchi, Shinji Watanabe, Nanxin Chen, Tetsuji Ogawa, and Tetsunori Kobayashi. Mask ctc: Non-autoregressive end-to-end asr with ctc and mask predict. *Proc. Interspeech 2020*, pages 3655–3659, 2020.

[12] Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al. A comparative study on transformer vs rnn in speech applications. In *ASRU*, pages 449–456. IEEE, 2019.

[13] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *EMNLP 2018*, page 66, 2018.

[14] Jinyu Li, Yu Wu, Yashesh Gaur, Chengyi Wang, Rui Zhao, and Shujie Liu. On the comparison of popular end-to-end models for large scale speech recognition. *Proc. Interspeech 2020*, pages 1–5, 2020.

[15] Christoph Lüscher, Eugen Beck, Kazuki Irie, Markus Kitza, Wilfried Michel, Albert Zeyer, Ralf Schlüter, and Hermann Ney. Rwth asr systems for librispeech: Hybrid vs attention–w/o data augmentation. *Interspeech*, pages 231–235, 2019.

[16] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *ICASSP*, pages 5206–5210. IEEE, 2015.

[17] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *Proc. Interspeech 2019*, pages 2613–2617, 2019.

[18] Kanishka Rao, Haşim Sak, and Rohit Prabhavalkar. Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer. In *ASRU*, pages 193–199, 2017.

[19] Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. Non-autoregressive machine translation with latent alignments. In *EMNLP*, pages 1098–1108, 2020.

[20] Hasim Sak, Matt Shannon, Kanishka Rao, and Françoise Beaufays. Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping. In *Interspeech*, volume 8, pages 1298–1302, 2017.

[21] Raphael Shu, Jason Lee, Hideki Nakayama, and Kyunghyun Cho. Latent-variable non-autoregressive neural machine translation with deterministic inference using a delta posterior. In *Proceedings of the AAAI*, volume 34, pages 8846–8853, 2020.

[22] Zhengkun Tian, Jiangyan Yi, Jianhua Tao, Ye Bai, Shuai Zhang, and Zhengqi Wen. Spike-triggered non-autoregressive transformer for end-to-end speech recognition. *Proc. Interspeech 2020*, pages 5026–5030, 2020.

[23] Ehsan Variani, David Rybach, Cyril Allauzen, and Michael Riley. Hybrid autoregressive transducer (hat). In *ICASSP*, pages 6139–6143. IEEE, 2020.

[24] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson-Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al. Espnet: End-to-end speech processing toolkit. *Proc. Interspeech 2018*, pages 2207–2211, 2018.

[25] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253, 2017.

[26] Albert Zeyer, André Merboldt, Ralf Schlüter, and Hermann Ney. A new training pipeline for an improved neural transducer. *Proc. Interspeech 2020*, pages 2812–2816, 2020.

[27] Pan Zhou, Ruchao Fan, Wei Chen, and Jia Jia. Improving generalization of transformer for speech recognition with parallel schedule sampling and relative positional embedding. *arXiv preprint arXiv:1911.00203*, 2019.