基于CTC对齐的端到端自动语音识别非自回归Transformer

Ruchao Fan，*IEEE* 学生会员，Wei Chu、Peng Chang 和 Abeer Alwan，*IEEE* 会士

arXiv:2304.07611v1 [cs.CL] 15 Apr 2023

摘要——近年来，端到端模型已广泛应用于自动语音识别（ASR）系统。最具代表性的两种方法是连接时序分类（CTC）和基于注意力的编码器-解码器（AED）模型。作为 AED 变体的自回归 Transformer 采用自回归机制生成词元，因此推理速度相对较慢。本文对一种用于端到端自动语音识别的基于 CTC 对齐的单步非自回归 Transformer（CASS-NAT）进行了全面研究。在 CASS-NAT 中，自回归 Transformer（AT）中的词嵌入被词元级声学嵌入（TAE）取代；TAE 利用 CTC 对齐提供的声学边界信息，从编码器输出中提取。TAE 可并行获取，从而实现输出词元的并行生成。在训练过程中，使用维特比对齐生成 TAE，并进一步探索多种训练策略，以提升词错误率（WER）表现。在推理过程中，深入研究了一种基于错误的对齐采样方法，以减少训练和测试过程中的对齐不匹配。实验结果表明，在各种自动语音识别任务中，CASS-NAT 的 WER 接近 AT，同时可实现 ∼24倍推理加速。无论是否采用自监督学习，我们都在多个数据集上取得了非自回归模型的最新成果。我们还分析了 CASS-NAT 解码器的行为，以解释其为何能达到与 AT 相近的表现。我们发现，TAE 在语法结构方面具有与词嵌入相似的功能，这可能表明即使不借助语言模型，也有可能从 TAE 中学习一些语义信息。

索引词—**CTC 对齐、非自回归Transformer、端到端自动语音识别、中间层损失。**

I. 引言

END-to-end模型在语音识别中表现出色，因为它们能够在同一个神经网络中同时实现声学模型、发音模型和语言模型的功能 [1], [2]。将上述组件联合训练可以减少中间错误，从而降低自动语音识别系统的词错误率（WER）。这种训练机制所需的模型参数也更少，因此适合端侧部署。连接主义时序分类（CTC） [3], 、基于注意力的编码器-解码器（AED） [4],和RNN转导器 [5], [6]是应用最广泛的端到端模型。采用最佳路径解码策略时，CTC的解码效率很高，但受限于其条件独立输出的假设。

R. Fan 和 A. Alwan 任职于美国加利福尼亚州洛杉矶市加州大学洛杉矶分校电气与计算机工程系，邮编 90095（电子邮件：fanruchao@g.ucla.edu, alwan@ee.ucla.edu）。W. Chu 和 P. Chang 任职于美国加利福尼亚州 PAII公司（电子邮件：{chuwei129,changpeng805}@pingan.com.cn）。本研究部分得到了美国国家科学基金会和加州大学洛杉矶分校-亚马逊科学中心的支持。

不过，AED和自回归 Transformer（AT） [7], [8], 模型通过引入语言模型式解码器来建模输出之间的依赖关系。然而，AT的解码采用自回归机制进行联合概率分解，导致输出词元逐步生成。这种机制会降低自动语音识别的推理速度，而推理速度是设计高效自动语音识别系统时至关重要的因素。

近年来，非自回归机制凭借并行生成输出词元的能力，提高了解码效率，因此受到越来越多的关注 [9]–[14]。Transformer 非自回归方法（NAT）主要分为两类：(i) 迭代式 NAT 和 (ii) 单步 NAT 或一次性 NAT。主流的迭代式 NAT 放宽了严格的非自回归条件，并通过K 次解码迭代生成输出。因此，迭代式 NAT 有时也称为“半非自回归”。而单步 NAT 可通过一次迭代生成输出序列。不同于神经机器翻译中将编码器输入扩展为解码器输入的方法，面向语音识别的单步 NAT 会提取高层声学表示作为解码器输入，并假设语言语义可由声学表示捕获 [15]–[17]。然而，这些声学表示要么是通过注意力机制隐式提取的 [15] ，要么仅利用 CTC 峰值而不够完整 [16],，这使得语言语义难以学习。

基于 CTC 的非自回归方法不断涌现，这得益于其高效率 [18]。例如，Chi 等人提出训练一个精炼器，根据精炼器先前的输出迭代改进 CTC 对齐 [19]。在[20], 中，每次迭代都会使用掩码词元作为解码器的先验信息，以增强 CTC 对齐。Nozaki 等人通过将中间预测用作额外输入，缓解了 CTC 输出相互独立的问题 [21]。此外，[22] 利用 Wav2vec2.0 预训练技术，大幅提升了纯 CTC 模型的词错误率表现。然而，这些方法的性能仍不及其自回归 Transformer（AT）对应方法。

本文通过利用 CTC 输出空间上的对齐，对一种非自回归框架进行了全面研究。该框架能够在一次迭代内生成输出序列，因此我们将其称为基于 CTC 对齐的单步 NAT（CASS-NAT）。CASS-NAT 包含四个主要模块：编码器、词元级声学嵌入提取器（TAEE）、自注意力解码器（SAD）和混合注意力

解码器（MAD）。编码器用于提取每一帧的高层声学表示。TAEE 利用 CTC 输出空间上的对齐所提供的信息，提取更有意义的词元级声学嵌入（TAE）。SAD 和 MAD 对 TAE 之间的依赖关系进行建模，其中 MAD 会直接利用编码器输出进行源注意力，而 SAD 则不会。不过，SAD 会通过 TAE 间接利用编码器的信息来执行自注意力。由于 TAE 可以并行获取，输出序列生成过程中不存在循环。同时，这两个解码器模块能够在潜在空间中对 TAE 之间的依赖关系进行建模。

我们将本文的贡献总结如下：1）通过详细实验考察各种解码器结构对词错误率的影响，而其中 SAD 和 MAD 的设置是在[23] 中凭直觉选定的；**2**）研究所提出的基于误差的采样对齐（ESA）方法中超参数的影响，例如采样阈值、采样对齐数量，以及用于对采样对齐进行排序的评分模型。研究揭示了准确率与推理效率之间的权衡，而 [23]中未涉及这一问题；**3**）本文比较了 [24] 中各项训练策略单独使用及其组合使用的效果，以便更好地理解所提出的训练策略。本文还纳入了 [24],中未涉及的知识蒸馏；**4**）鉴于 CTC 对齐的质量与模型准确率密切相关，本文研究了用于 CASS-NAT 训练的多种编码器初始化方案（包括 AT编码器、CTC编码器和随机初始化）。HuBERT编码器 [25]也纳入了比较。早期发表的工作并未进行此类研究；以及**5**）在多种数据集上应用所提出的方法（LibriSpeech：成人朗读语音；TED2：成人自发语音；MyST：儿童自发语音；Aishell1：成人普通话朗读语音），以验证算法的泛化能力，并为非自回归模型取得新的自动语音识别最优结果。我们早期发表的工作仅报告了 LibriSpeech 和 Aishell1 上的结果。

本文其余部分组织如下。第二节介绍端到端模型的背景及相关工作。第三节介绍 CASS-NAT 框架，包括系统架构以及训练和推理策略。第四节介绍实验设置，第五节展示并讨论实验结果。第六节总结全文。

II. 背景

我们首先回顾所提出方法背后的重要概念，并介绍基本符号。令 $X=(x_1, ..., x_t, ..., x_T)$ 表示输入序列，其中 x_t 包含第 t 帧的语音特征。$Y=(y_1, ..., y_u, ..., y_U)$ 是输出序列，其中 y_u 是位置 u 处的词元。语音识别任务的目标是根据声学信息 X 找到最可能的转录文本 Y，可表示为 $Y^*=\text{argmax}\ P(Y|X)$。

A. 自回归模型

近年来，Transformer 被证明是性能最佳的自回归模型（AT） [8], [26]。AT 采用编码器-解码器结构，解码器根据所有先前词元生成每个词元。这种架构设计通过一系列条件概率实现序列建模，其中每个条件概率都构成一个分类问题。随后，AT 模型通过如下目标函数进行训练：

$$\begin{aligned} L_{AT} &= -\log\ P(Y|X) = -\log \prod_{i=1}^{U} P(y_i|y_{<i}, X) \\ &= -\sum_{i=1}^{U} \log\ P(y_i|y_{<i}, X) \end{aligned} \tag{1}$$

其中，P 是自回归模型的概率分布，$y_{<i}$ 是位于 i^{th} 词元之前的所有词元。

自回归模型可以通过将所有真实词元用作解码器输入（教师强制）或使用并行计划采样 [27] 来高效训练。然而，在推理期间，需要使用集束搜索算法在搜索空间中获取概率最高的序列。集束搜索以及生成时必须使用历史词元的要求，破坏了自回归模型的并行性，导致推理速度较慢。

B. 非自回归模型

非自回归模型中的词元之间没有严格的依赖关系。例如，CTC采用条件独立假设，而Mask-CTC [14] 则将解码器训练为掩码语言模型，从而在被掩码和未被掩码的词元之间建立较弱的依赖关系。通过放宽依赖假设，可以并行生成所有词元，从而提高推理速度。

1）*CTC*及其对齐：CTC 模型中的词元之间没有依赖假设。给定 X 时，Y 的后验概率可以直接通过将每一帧对应的概率相乘来计算。设$Z=\{z_1, ..., z_t, ..., z_T\}$ 为模型的输出，其中 z_t表示对应于输入x_t的时间步 t 上的输出。然而，对于语音识别任务，Z的长度总是大于 Y的长度。为了计算损失，词汇表中会添加一个特殊的空白词元 b；b 可以在任意时间步被预测为 Z 的输出。推理时，移除 b 和重复词元，以获得较短的序列 Y，此过程可定义为映射规则 β。训练期间，存在多个 Z，可依据规则 β映射为 Y。因此，CTC损失是所有此类 Z的总和，公式如下 [3]:

$$\begin{aligned} L_{CTC} &= -\log\ P(Y|X) = -\log \sum_{Z\in\beta^{-1}(Y)} P(Z|X) \\ &= -\log \sum_{Z\in\beta^{-1}(Y)} \prod_{t=1}^{T} P(z_t|X) \end{aligned} \tag{2}$$

其中 β^{-1} 是映射规则的逆。

每个 Z中都呈现了 X 中所有时间步与 Y 中词元之间的对齐关系。因此，Z 也称为

CTC 对齐，这与基于 HMM 的自动语音识别系统中的对齐类似。本文探讨了利用对齐 Z 所提供的信息，实现端到端语音识别的单步非自回归模型的优势。当 Z^* 映射到真实标注 Y 的概率最高时，它被称为维特比对齐。

2) 单步非自回归：单步 NAT 采用编码器-解码器结构，使输出长度与 Y 相同。输出词元之间仍然条件独立，这与 CTC 类似。但这种方法隐含地假设语言语义可以由高级声学表征捕获，而这些表征类似于词嵌入。模型针对每个词元使用交叉熵损失进行更新，其形式如下：

$$L_{NAT} = -\log P(Y|X) = -\sum_{i=1}^{U} \log P(y_i|X) \tag{3}$$

C. 其他相关工作

1) 卷积增强自注意力：Transformer 中的自注意力模块通过对整个序列进行加权求和来捕获全局信息。然而，局部信息对序列建模也很重要。以语音特征为例，每个元音或辅音中的频率细节有助于识别这些声音。在计算机视觉领域，卷积层已被证明能够很好地捕获卷积核 [28] 范围内的局部细节。近期的研究在自动语音识别中采用这一思路，通过卷积模块 [29]–[31] 增强 Transformer。每个自注意力模块中也使用了相对位置编码。卷积增强自注意力块能够有效提升性能，但只有应用于 AT 编码器时，提升才显著。另一方面，AT 解码器采用因果结构（注意力使用上三角掩码矩阵），借助卷积模块捕获的局部细节较少，因此这种情况下性能提升并不显著。本文除了在编码器中使用卷积增强自注意力层外，还探索了将其用于非自回归翻译解码器。

2) 中间层损失：深层 Transformer 总是会遭遇梯度消失，尤其是距离输出层较远的参数。先前提出的中间层损失是在每一层之后添加额外的损失函数，以促进梯度更新 [32], [33]。已有研究证明，使用中间 CTC 损失有助于提升每个-
CTC 模型的性能 [34], [35]。在 [36],中，中间交叉熵损失被用于训练基于Transformer的深度声学模型，以构建基于HMM的混合自动语音识别系统。本文在所提出的框架中结合使用中间 CTC 和交叉熵损失。

3) 自监督学习：近年来，自监督学习（SSL）广受欢迎，尤其适用于低资源任务 [25], [37]。我们在实验中使用 HuBERT 进一步提升系统性能。在预训练期间，HuBERT 模型根据未掩码区域重建掩码输入。我们利用自监督预训练来提升编码器的建模能力，进而提高 CTC

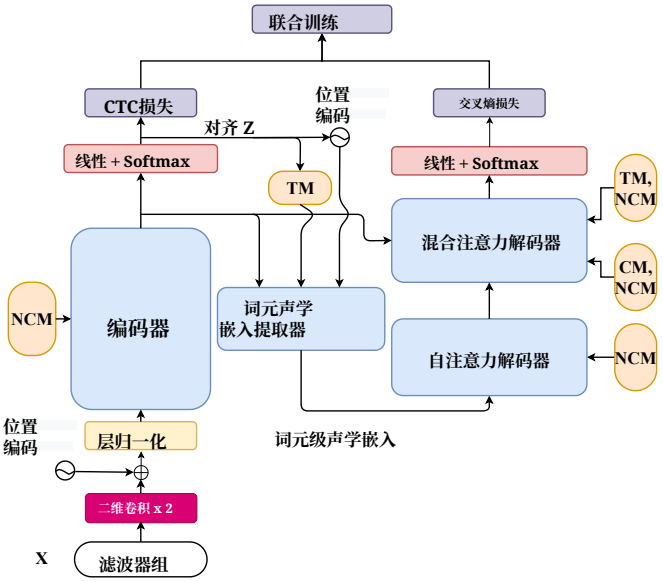


图1：所提出的 CASS-NAT 架构概览。CM 和 NCM 分别表示因果掩码和非因果掩码。TM 表示触发掩码。

对齐的质量，这对于非自回归翻译解码器中准确的语言语义建模至关重要。

III. 所提出的框架：CASS-NAT

本节介绍所提出的基于 CTC 对齐的单步非自回归 Transformer（CASS-NAT）。我们将阐述目标函数的数学推导、用于提升性能的训练策略，以及所提出的基于采样的解码策略。

A. 系统架构

所提出的 CASS-NAT 系统架构基于 CTC/注意力混合架构 [38]，利用 CTC对齐实现非自回归。图 1 展示了 CASS-NAT 的四个主要模块：编码器、词元声学嵌入提取器（TAEE）、自注意力解码器（SAD）和混合注意力解码器（MAD）。

*1)*注意力中的掩码：注意力机制对于 Transformer 至关重要。注意力机制中最基本的计算是以序列为输入的缩放点积自注意力。传统的自注意力会利用整个序列中的信息。但自注意力机制的每个输出不一定都依赖于整个输入序列。例如，神经语言模型会使用上三角矩阵（掩码矩阵），仅收集序列中过去词元的信息。图 1 展示了四个主要模块中用于不同目的的三种掩码矩阵。目前我们不考虑流式自动语音识别模型，因此编码器中使用非因果掩码（NCM）。NCM 是一种将填充部分置为零的矩阵，以防止填充词元或填充帧参与注意力计算。在 TAEE 中，使用触发掩码（TM）来提取准确的词元级声学嵌入。TM 会标记出被触发的帧，使所用帧的位置标记为 1，而

其他位置标记为 0。触发掩码的示例见 III-A3节。MAD 包含自注意力层和交叉注意力层，与 AT 解码器类似。在这种情况下，自注意力层可以使用因果掩码（CM）或 NCM，而交叉注意力层可以使用 NCM 或 TM。第五节-A 对 MAD 中掩码矩阵的不同选择进行了实验探索。最终，可以通过掩码矩阵对自注意力计算进行扩展，如下所示：

$$Attention(Q,K,V,M)=\left(Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)\otimes M\right)V \tag{4}$$

其中，$Q \in R^{n_q\times d_q}$、$K \in R^{n_k\times d_k}$、$V \in R^{n_v\times d_v}$ 和$M \in R^{n_q\times n_k}$ 分别是查询、键、值和掩码矩阵。

2) 编码器：编码器从语音特征 X中提取高层声学表示 H 。如图1所示，在编码器之后添加了一个带有CTC 损失函数的线性层。CTC的作用是在CTC 输出空间上获得对齐，为词元声学嵌入提取器（TAEE）提供辅助信息。训练期间使用维特比对齐。推理期间，我们通过实验探索了从CTC 输出空间进行采样的各种方法。

*3)*词元声学嵌入提取器：词元声学嵌入提取器旨在利用CTC 对齐提供的辅助信息，提取词元级声学嵌入（TAE）。例如，给定一个对齐$Z=\{z_1,...,z_t,...,z_T\}$，我们可以为每个词元 u 估算一个声学片段 $\{t_{u-1}+1,...,t_u\}$（注意，此处的1表示一帧），以及 Z中的词元数量。

首先，CTC对齐为每个词元提供一个声学边界 y_u，随后将其转换为触发掩码。具体而言，我们定义了一个从对齐到触发掩码的映射规则，并在训练和推理阶段固定该规则。我们将对齐中每个词元对应的第一个非空白索引视为其结束边界。其直觉依据是我们的假设：模型只有在获取该词元的全部声学信息后，才会输出该词元。使用第一个非空白索引是为了简化处理，并确保训练和解码的一致性。例如，若真实标注 $Y=\{C,A,T\}$ 的对齐为 $Z=\{_,C,C,_,A,_,_,T,_\}$，其中 代表空白符号，则 C 和 A 的结束边界分别为 Z_2 和 Z_5，因此词元 A 的触发掩码为 $[0,0,1,1,1,0,0,0,0]$。该映射规则对声学分段的划分可能并不准确，但在训练和解码期间应保持一致。此处的触发掩码不同于 [39] 中为流式处理目的所使用的触发掩码。在[39], 中，先前的声学表示可以被每个词元重复使用，因此前述示例中词元 A 的触发掩码为 $[1,1,1,1,1,0,0,0,0]$。

其次，CTC 对齐为解码器输入提供词元数量。移除空白符号和重复项后，对齐中的词元数量 Z 用作正弦位置编码的预测长度（解码器输入长度）。如图 1 所示，随后使用触发掩码和正弦位置编码，通过单层源注意力块提取每个词元的 TAE。

TAE 取代自回归模型中的词嵌入，以实现每个序列的并行生成。

4) 自注意力解码器：从 TAEE 中提取的 TAE（见 III-A3 节）具有并行生成这一良好特性，因此被用作解码器输入的词嵌入的替代。由于无需在解码器中建立循环，我们在自注意力解码器（SAD）中使用非因果掩码（NCM）来建模 TAE 之间的关系。

5) 混合注意力解码器：我们假设 TAE 与词嵌入具有相似的学习语言语义能力。因此，我们设计了一种混合注意力解码器（MAD），以重新利用编码器信息，从而在输出层做出更好的决策。与 AT 解码器类似，MAD 包含一个使用因果掩码（CM）或非因果掩码（NCM）的自注意力层，以及一个使用触发掩码（TM）或非因果掩码（NCM）的源注意力层。在 MAD 后添加一个线性层，随后计算交叉熵损失。由于训练期间使用了维特比对齐，输出长度与真实标注 Y 相同。

B. 训练细节

本节介绍训练准则，随后介绍用于提升 CASS-NAT 性能的各种训练策略。

1) 训练准则：在我们的框架中，CTC 对齐Z 被引入为潜变量。给定第 II 节中的 X 和 Y，对数后验概率可分解为：

$$\log P(Y|X)=\log \mathbb{E}_{Z|X}[P(Y|Z,X)],\quad Z\in q. \tag{5}$$

其中，q 是可映射到 Y的对齐集合。对于不属于 q的对齐，我们假设$P(Y|Z,X)$等于零。为降低计算成本，我们采用最大近似 [40]:

$$\begin{aligned}\log P(Y|X) &\geq \mathbb{E}_{Z|X}[\log P(Y|Z,X)]\\ &\approx \max_Z \log \prod_{u=1}^{U} P(y_u|z_{t_{u-1}+1:t_u},x_{1:T})\end{aligned} \tag{6}$$

其中，$\mathbb{E}$ 表示期望，t_u 是词元 u $(t_0=0)$的结束边界。所有 t_u均可根据对齐 Z估计。我们认为，TAE 在一定程度上能够捕获语言语义（分析见第 V-E 节），这有助于缓解输出词元之间的独立性导致的性能下降。

该框架通过联合优化编码器侧的 CTC 损失函数（L_{CTC}）和解码器侧的交叉熵（CE）损失函数（L_{dec}）进行训练，任务比率为 λ[38]，因此最终损失函数（L_{joint}）定义为：

$$L_{\text{joint}}=-\lambda\cdot\log\sum_{Z\in q}\prod_{i=1}^{T}P(z_i|X)-\log\prod_{u=1}^{U}P(y_u|z^*_{t_{u-1}+1:t_u},X) \tag{7}$$

其中，P 是概率分布，Z^* 是最可能的对齐（维特比对齐）。第二项是根据公式 6 计算的对数后验概率的最大近似。

2) 卷积增强自注意力块：公式 4 中的自注意力计算考虑了序列中的全局信息，但忽略了局部细节。为了解决这一问题，我们提出卷积增强自注意力块，以强调对编码器中输入序列局部依赖关系的建模 [30], [41]。与先前的工作不同，我们也在 SAD 和 MAD 中使用卷积增强自注意力块。具体而言，前馈层被拆分为两个子层，分别置于块的开头和结尾。在自注意力层之后插入一个与 [30] 中类似的卷积层，唯一不同的是，根据经验，我们使用层归一化而非批归一化。 i^{th}MAD 中的最终计算可表示为：

$$\hat{s_i} = s_i + \frac{1}{2}\text{FFN}(s_i) \tag{8}$$

$$s_i^{'} = \hat{s_i} + \text{LN}(\text{Attn}(\hat{s_i}, \hat{s_i}, \hat{s_i}, \text{NCM})) \tag{9}$$

$$s_i^{''} = s_i^{'} + \text{Conv}(s_i^{'}) \tag{10}$$

$$s_i^{'''} = s_i^{''} + \text{LN}(\text{Attn}(s_i^{''}, H, H, \text{NCM})) \tag{11}$$

$$o_i = \text{LN}(s_i^{'''} + \frac{1}{2}\text{FFN}(s_i^{'''})) \tag{12}$$

其中，LN 表示层归一化，FFN 表示前馈网络。NCM 是非因果掩码。 s_i 和 o_i 分别是模块 i的输入和输出。卷积增强型自注意力模块也可用于其他非自回归模型。

不同于相对位置编码在 [30], [42], 中的用法，我们采用 [27]中的设定，将相对位置的最大长度设为 k。因此，学习 $2k+1$ 位置嵌入，以表示 $[-k, k]$之间的相对位置。

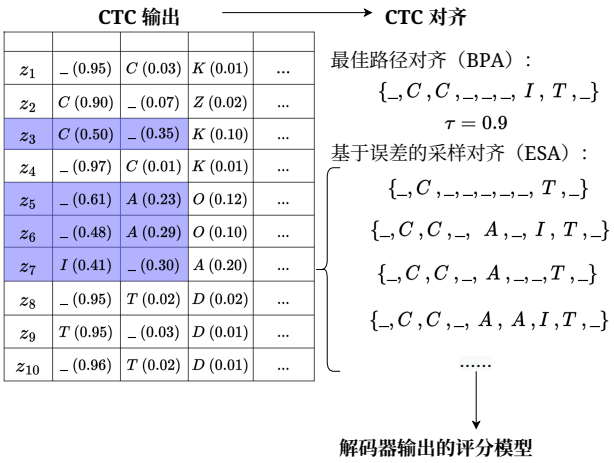
*3)*中间层损失：由于 CTC 和交叉熵损失函数在 CASS-NAT 框架中联合优化，我们将中间 CTC 和中间交叉熵损失函数纳入式 7，使不同层的参数能够以相同的尺度更新。令 $L_{dec} = -\log \prod_{u=1}^{U} P(y_u|z^*_{t_{u-1}+1:\ t_u},\ X)$ 和 $L_{CTC} = -\log \sum_{Z\in q} \prod_{i=1}^{T} P(z_i|X)$，则目标函数可改写为：

$$\begin{aligned} L_{\text{joint}} = {} & \lambda_{CE} L_{dec}^{final} + (1-\lambda_{CE}) L_{dec}^{mid} \\ & + \lambda_{CTC} L_{CTC}^{final} + (1-\lambda_{CTC}) L_{CTC}^{mid} \end{aligned} \tag{13}$$

其中 λ_{CE}和 λ_{CTC}是任务比率。 mid和 $final$表示插入损失函数的层位置。我们发现，与自回归模型相比，中间层损失对 CASS-NAT 更有效 [24]。其他非自回归模型也可以添加中间层损失。

4) 触发掩码扩展：TAE 的质量取决于触发掩码（TM）的准确率，该掩码由 CTC 对齐映射而来。尽管使用 CTC 损失函数来优化对齐，但在进行强制

在 CTC 输出空间上进行对齐，导致 TM 不准确。为了解决这个问题，我们扩展 TM，使其包含每个词元的上下文帧。例如，假设上下文帧大小为一，则词元 U 的声学边界变为$\{z_{t_{u-1}}, ..., z_{t_u+1}\}$。随后将扩展触发掩码



z_1	_ (0.95)	C (0.03)	K (0.01)	...
z_2	C (0.90)	_ (0.07)	Z (0.02)	...
z_3	C (0.50)	_ (0.35)	K (0.10)	...
z_4	_ (0.97)	C (0.01)	K (0.01)	...
z_5	_ (0.61)	A (0.23)	O (0.12)	...
z_6	_ (0.48)	A (0.29)	O (0.10)	...
z_7	I (0.41)	_ (0.30)	A (0.20)	...
z_8	_ (0.95)	T (0.02)	D (0.02)	...
z_9	T (0.95)	_ (0.03)	D (0.01)	...
z_{10}	_ (0.96)	T (0.02)	D (0.01)	...

图2：从 CTC 输出中获得 CTC 对齐的示意图，包括最佳路径对齐（BPA）和基于错误的采样对齐（ESA）。 C (0.90) 表示$P(z_i = C|X) = 0.90$。“ ”表示空白词元。采样的阈值 τ 设为 0.9。

一帧，以用于后续的声学嵌入提取。请注意，触发掩码扩展是专为 CASS-NAT 设计的。

C. 推理：基于错误的采样解码

解码时，关键是获得与训练中使用的假设维特比对齐接近的 CTC 对齐。然而，转录文本并不可用。我们提出了三种不同的推理阶段对齐生成方法：（1）最佳路径对齐（BPA）；（2）束搜索对齐（BSA）；以及（3）基于错误的采样对齐（ESA）。我们还展示了将维特比对齐用作词错误率上限的结果，假设解码时可以获取转录文本，这种方法称为预言机对齐。**BPA** 类似于 CTC 贪心解码：它在每个时间步选择概率最高的词元，但不会在最终序列中移除空白词元和重复标记。**BSA** 类似于在 CTC 输出空间上进行束搜索解码，得到解码过程中概率最高的对齐。与 BPA 相比，BSA 理应生成更接近预言机对齐的对齐，从而降低词错误率，但这会破坏 CASS-NAT 的并行性，导致实时因子（RTF）显著增加。

考虑到式（6）中的期望，我们提出第三种对齐生成方法：根据 BPA 中可能出现的错误进行采样。该方法称为基于误差的采样对齐（ESA）。生成 ESA 时，阈值 τ 用于决定是否在每个时间步进行采样。如果输出分布的最高概率大于τ，则遵循 BPA 规则；否则，从概率最高的前两个词元中随机采样。如图2中蓝色阴影部分所示，由于前 1 个词元的概率较低， z_3、 z_5、 z_6 和 z_7 处的 CTC 输出需要进行采样。根据所提出的采样规则，图2右侧展示了四种采样对齐作为示例。

在前两个词元中进行采样，是因为触发掩码对空白标记较为敏感，而 BPA 中大多数错误输出的前两个词元中都包含空白标记。此外，在两个词元中进行采样效率较高，因为采样空间较小。ESA 旨在纠正容易出错的输出。在解码阶段进行采样不会对推理速度产生太大影响，因为采样可以并行完成。ESA 生成的对齐可能比 BPA 更接近预言机对齐。与 BSA 相比，ESA 可以并行实现，不会增加实时因子（RTF）。最后，可以使用自回归模型基线或语言模型对对齐进行评分，并找出整体最佳对齐。请注意，与 BPA 相比，ESA 的解码器输入长度可能不同。如图2所示，BPA 的解码器长度为 4（包括一个序列结束词元），而 ESA 的长度分别为 3、5、4 和 5。词元数量的这种变化使 ESA 有可能采样得到与预言机对齐长度相同的对齐，这对非自回归模型的性能至关重要，实验结果将对此进行验证。请注意，ESA 解码专为 CASS-NAT 提出。

IV. 实验设置

A. 数据准备

为考察所提出框架的有效性，我们开展了若干自动语音识别任务，涵盖朗读语音与自发语音、英语与普通话语音，以及成人与儿童语音。我们选取了四个数据集：（1）包含朗读语音的 960 小时 LibriSpeech 英语语料库 [43]；（2）包含成人朗读语音的 178 小时 Aishell1 普通话语料库 [44]；（3）包含 TED 演讲语音的 210 小时 TEDLIUM2（TED2）英语语料库 [45]；（4）包含自发语音的 My Science Tutor（MyST）儿童英语语料库 [46]。我们使用 MyST 中已标注部分，占该语料库的 42%（240 小时）。

所有实验均使用 80 维对数梅尔滤波器组特征，每 10 毫秒计算一次，采用 25 毫秒汉明窗。将每 3 个连续帧的特征拼接，形成 240 维特征向量作为输入。输出标签集合由 LibriSpeech 的 5k 个子词单元，以及 TED2 和 MyST 各自的 500 个子词单元组成。所有子词单元均使用各数据集的训练集，通过 SentencePiece [47] 生成。Aishell1 数据集使用 4230 个汉字作为词汇表。为避免过拟合，我们对 TED2、Aishell1 和 MyST 的滤波器组特征应用了速度扰动[48] 和 SpecAugment [49]。由于计算资源有限，LibriSpeech 未使用速度扰动。

B. 网络架构

1) 自回归模型：首先针对 LibriSpeech 使用架构（$N_e = 12$, $N_d = 6$,$d_{ff} = 2048$, $nh = 8$, $d_{att} = 512$）训练一个 CTC/注意力 AT 基线，针对其他三个数据集则使用架构（$N_e = 12$, $N_d = 6$, $d_{ff} = 2048$, $nh = 4$, $d_{att} = 256$）进行训练，其中 d_{ff} 表示前馈神经网络模块的维度，d_{att} 表示注意力模块的维度，nh 表示注意力头的数量，N_e 和 N_d 分别表示编码器和解码器块的数量。在编码器之前，采用两个卷积层，每层包含 64 个滤波器，卷积核大小为 3，步幅为 2，从而使帧率降低 4 倍。将 Conformer结构用于 AT编码器时，我们将 d_{ff} 缩减至 1024，以使模型的参数数量与 Transformer基线相同，并将相对位置 k 的最大长度设为 20。卷积模块中的卷积核大小在 LibriSpeech 上为 31，在其他三个数据集上为 15。

2) CASS-NAT：训练期间，为加快收敛，CASS-NAT编码器使用AT编码器进行初始化，如 [50]所示。AT基线中的解码器被替换为1层TAEE、m层SAD和n层MAD。我们使用LibriSpeech数据集对 m 和 n 进行研究，并将最佳设置应用于其他三个数据集。对于卷积增强解码器，前馈层的维度也减半；对于使用子词单元的任务，相对位置的最大长度设为8，而对于Aishell1数据，该值设为4。触发掩码扩展的上下文帧设为1，因为我们发现进一步增大扩展范围并无额外改进。中间损失函数被插入编码器和MAD的中间层，其中 λ_{CE} 为0.99，λ_{CTC} 为0.5。推理期间会移除插入的投影层。这些设置是通过经验确定的。

C. 训练与解码设置

所有实验均使用 PyTorch [51] [1] 实现。LibriSpeech 和 Aishell 数据集采用 [49] 中的 Double 调度策略作为学习率调度策略：学习率先升高至 0.001 并保持不变，随后按指数衰减至 1e-5。TED2 和 MyST 数据集采用 [7] 中基于 Transformer的调度器，预热步数为 25k，Noam因子为 5。层归一化、丢弃率为 0.1 的 dropout，以及惩罚系数为 0.1 的标签平滑，均作为训练 Transformer 的通用策略。我们计算开发集的词错误率，以进行提前停止（连续 11 个训练轮次无改善时停止）。最终评估使用最后 12 个训练轮次的平均结果。大多数实验在 90 个训练轮次内完成。为研究 ESA 中不同模型对对齐进行评分的影响，我们使用 LibriSpeech 提供的文本训练了一个基于Transformer的语言模型。实验还比较了数据集中提供的 n 元语法模型。对于预训练HuBERT编码器，我们使用在 LibriSpeech 960 小时语料库上训练的 Fairseq 模型微调英语数据模型，并使用在 10000 小时 WenetSpeech[52] 上训练的 Tecent模型微调普通话数据模型。

在 AT 解码过程中，LibriSpeech 的束宽设为 20，其他三个数据集则设为 10。束搜索解码过程中未使用外部语言模型。实时因子（RTF）的评估在批大小为 1 的情况下，使用 V100 GPU 进行。对于使用 ESA 的 CASS-NAT 解码，阈值、采样对齐数量和评分模型将在第 V-B 节中进行探讨。

V. 结果与分析

第一组自动语音识别实验使用 LibriSpeech数据集，探究了不同的解码器结构及其影响因素

[1]我们的代码将发布在 https://github.com/Diamondfan/cassnat asr_

表 I：使用 LibriSpeech 时，自注意力解码器（SAD）和混合注意力解码器（MAD）在不同块数下的词错误率。还探究了 MAD 中使用的各种掩码矩阵，其中 CM、NCM 和 TM 分别表示因果掩码、非因果掩码和触发掩码。例如，NCM +TM 表示 MAD 的自注意力采用 NCM，源注意力采用 TM。粗体数字表示最佳结果。

解码器结构	掩码在 MAD	dev-干净	dev-其他	测试集-干净	测试集-其他
7SAD + 0MAD	-	5.3	11.1	5.4	11.1
5SAD + 2MAD	NCM + NCM	**4.7**	**10.4**	**4.8**	**10.3**
	NCM +触发掩码	4.7	10.5	4.9	10.5
	因果掩码 + NCM	4.8	10.7	4.9	10.6
3SAD + 4MAD	NCM + NCM	4.7	10.4	4.8	10.4
1SAD + 6MAD	NCM + NCM	4.6	10.5	4.7	10.4

在 ESA 解码、CTC 对齐行为以及所提出训练策略的有效性方面。采用在 LibriSpeech 任务中确定的最佳设置，在另外三个数据集上进行实验。

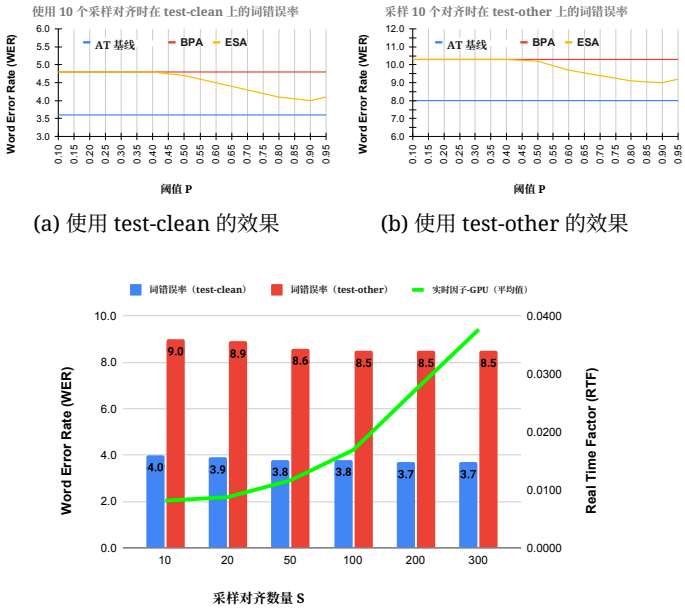
A. CASS-NAT 解码器的结构

为研究 CASS-NAT 解码器的结构，我们在解码器中采用 m 个 SAD 模块和 n 个 MAD 模块的不同组合（$m+n=7$），以使模型参数数量与 AT 基线相近。推理时采用最佳路径对齐，并将在下一节讨论其他解码策略。结果如表 I 所示。如表中所示，不使用 MAD（第一行中的 0 MAD）会导致性能相比其他配置大幅下降，这表明词元级语音表示可能需要重新获取细粒度的帧级信息（编码器输出），才能更好地进行上下文建模。使用 5 个 SAD 模块和 2 个 MAD 模块时性能最佳（在 test-other 集上的词错误率为 10.3%）。

由于每个 MAD 中有两个注意力层（自注意力和源注意力），我们针对这两个层尝试了不同的掩码矩阵。对于自注意力层，我们考虑使用因果掩码（CM）或非因果掩码（NCM）；对于源注意力层，则使用触发掩码（TM）或 NCM。表 I 中的结果表明，在两个注意力层上都使用 NCM 时，词错误率最低。尽管 TAE 被视为词嵌入的替代方案，但对于 CASS-NAT 解码器中词元级声学嵌入的上下文建模而言，CM 似乎并非必需。本节中性能最佳的设置（5 个 SAD 和
在两个注意力层中均采用带有 NCM 的 2 个 MAD）作为后续实验的默认设置。

B. 基于误差的采样对齐（*ESA*）解码

训练时使用维特比对齐，但推理时无法获得。为减少训练与推理之间的对齐不匹配，我们提出了基于错误的采样对齐方法（ESA）。有三个重要因素会影响 ESA 的词错误率表现：采样阈值 P、采样对齐数量 S，以及用于



(a) 使用 test-clean 的效果 (b) 使用 test-other 的效果

(c) 采样对齐数量的影响

图 3：基于误差的采样对齐（ESA）解码中不同阈值P 和采样对齐数量 S 下的词错误率表现。使用一块 V100 GPU、批大小为 1 时评估实时因子（RTF）。

对对齐进行排序的评分模型。我们使用上一节中最优的解码器结构（5SAD + 2MAD），评估不同配置下 ESA 解码的表现。

图 3a 和图 3b 分别展示了在 LibriSpeech test-clean 和 test-other 数据集上，阈值P 从 0.10 变化到 0.95 时的结果。此处采样对齐数量为 10，评分模型为 AT 基线。我们还列出了 AT 基线和采用最佳路径对齐（BPA）的 CASS-NAT 解码的词错误率，以作比较。如图所示，当 $P = 0.9$时，ESA 的性能最佳。较高的阈值意味着采样对齐数量较少，因此无法观察到进一步的提升。后续实验中，阈值 P 设为 0.9。

图 3c 展示了采样对齐数量 S 对测试集 test-clean 和 test-other 上词错误率和实时因子的影响。如图所示，随着采样对齐数量增加，ESA 解码的词错误率有所改善，但当 S 大于 50 时，提升幅度较小。同时，实时因子会随着 S 增加而迅速上升。总体而言，后续实验中将 $S = 50$ 用作默认值可能最为合适。

最后，我们比较了各种评分模型。我们考虑使用 AT基线、神经语言模型（NLM）和 n元语法模型对采样对齐进行排序。NLM 是一种基于 Transformer 的模型，使用 LibriSpeech语料库中提供的文本进行训练。n元语法模型同样来自 LibriSpeech语料库。ESA解码以及 BPA 和 ESA解码的结果如表II所示。正如第 III-C 节所分析的，BPA 具有显著的加速比，但其词错误率远高于

表 II：从词错误率和图形处理器加速比方面比较 BPA、BSA 和 ESA解码。NLM 是基于 Transformer 的神经语言模型。三元语法模型和四元语法模型均来自 LibriSpeech。

	评分模型	dev-干净	dev-其他	测试集-干净	测试集-其他	GPU加速比
AT基线	-	3.4	8.1	3.6	8.0	1.00倍
Oracle	-	2.1	5.5	2.2	5.3	39.6x
BPA	-	4.7	10.4	4.8	10.3	90.0x
BSA	-	3.8	8.8	3.9	8.8	0.90x
ESA	AT基线	3.6	8.8	3.8	8.6	28.4x
	NLM	3.6	8.9	3.9	8.7	31.6倍
	三元语法模型	5.4	11.4	5.8	11.4	32.6倍
	四元语法模型	5.4	11.3	5.7	11.3	31.5倍

AT基线，而BSA能取得更好的词错误率（WER），但速度甚至比AT基线更慢。通过使用基于神经网络的评分模型，ESA可以兼具BPA和BSA的优势。例如，使用NLM作为评分模型时，ESA在LibriSpeech测试集其他数据上可实现8.7%的词错误率和31.6倍的加速比。与BPA相比，加速比有所下降，这是排序过程造成的。使用n元语法模型对齐进行排序时，WER比ESA中使用NLM更差，因为n元语法模型的语言建模能力不如NLM。另一个可能的原因是，n元语法模型的概率分布与CASS-NAT解码器的输出不同，而NLM的概率分布则与之相似。n元语法模型的图形处理器加速比与NLM相近，原因在于无法同时获取采样对齐的分数。性能最佳的评分模型是AT基线，因此在后续实验中，ESA解码默认使用AT基线。

请注意，由于 ESA 解码中的采样过程，随机数生成器的不同种子可能会导致词错误率略有差异。幸运的是，在我们的实验中，随机性较小，相对方差约为 0.5%。

C. ESA 为何有效？——*CTC* 对齐分析

本节分析通过不同解码策略获得的 CTC 对齐，以了解 ESA 为何能够提升 CASS-NAT 的性能。我们在 LibriSpeech 测试集上评估了两项指标：失配率（MR）和长度预测错误率（LPER）。MR 和 LPER 是根据解码时使用的对齐与预言机对齐计算的，其中移除了空白标记和重复标记。MR 是相较于预言机对齐的删除错误和插入错误之比；由于替换错误不会改变声学

每个词元的边界或预测词元的数量。如果某个对齐中的词元数量与预言机对齐中的数量不同，则该对齐被视为长度预测错误。LPER 是存在长度预测错误的话语所占的百分比。

MR 和 LPER 的结果如表 III 所示。请注意，在假设解码过程中可以获得转录文本的情况下，使用预言机对齐时，test-clean 数据集上的词错误率下限为 2.2%；这表明该框架具有良好的应用前景。在比较不同解码策略的这两个指标时，我们发现，词错误率与 LPER 的相关性高于与 MR 的相关性。这

表 III：CASS-非自回归解码的不同解码方法比较。**Oracle**：使用真实标注的维特比对齐。**MR**：失配率；**LPER**：长度预测错误率（以子词为建模单元）。**S**：ESA 中采样的对齐数量。

解码方法	S	词错误率 (%)		失配率 (%)		长度预测错误率 (%)	
		测试集-干净	测试集-其他	测试集-干净	测试集-其他	测试集-干净	测试集-其他
Oracle	1	2.2	5.3	-	-	-	-
BSA	1	4.0	9.2	2.5	6.0	28.09	48.83
BPA	1	4.8	10.3	2.4	5.2	34.92	51.68
ESA	10	4.0	9.0	3.6	6.0	26.41	44.91
	50	3.8	8.6	3.8	6.3	25.46	43.08
	100	3.8	8.5	3.8	6.2	25.61	42.70
	300	3.7	8.5	3.8	6.3	25.53	42.67

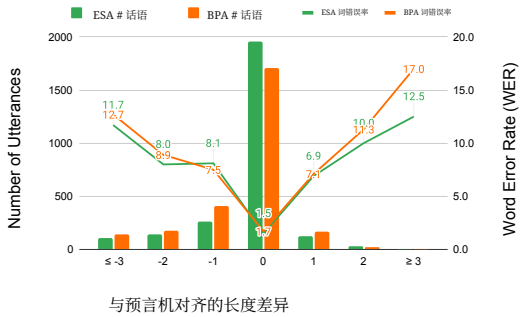


图 4：在测试集干净数据集上使用 ESA(s=50) 解码时的长度预测错误分布及其对应的词错误率。

表明，准确估计输出长度（即解码器输入的长度）对于非自回归模型非常重要， [13] 中也提到了这一点。其原因可能是，当长度预测准确时，CASS-NAT 解码器能够通过基于上下文的词元级声学建模，更强地纠正 CTC 对齐中的错误。为进一步验证这一假设，我们在图 4 中绘制了长度预测错误的分布及其对应的词错误率。如图所示，当解码器输入的长度估计准确时，CASS-NAT 的词错误率低于 2%；而对于那些与预言机对齐相比长度绝对差值超过 3 的话语，其词错误率高于 10%。

D. 改进 *CASS-NAT* 的训练

尽管采用了 ESA 解码方法，CASS-NAT 的词错误率表现（8.6%）仍不如 AT基线（8.0%）。本节尝试通过采用多种训练策略来提升 CASS-NAT 的词错误率表现，这些策略包括卷积增强编码器（ConvEnc.）、中间 CTC 损失（InterCTC）、卷积增强解码器（ConvDec.）、中间交叉熵损失（In- terCE）和触发掩码扩展（Mask Exp.）。表 IV 展示了不同训练策略组合的结果。

首先，我们将 ConvEnc. 和 InterCTC 应用于自回归 Transformer，构建 newAT基线，以便进行公平比较。请注意，我们已将 ConvDec. 和 InterCE 应用于

表 IV：在 LibriSpeech 数据上采用所提出训练策略时的词错误率。所有配置均使用 SpecAug。WERR 表示相较于各自对应基线系统，在 test-other 测试集上的相对词错误率改善幅度。KD 表示利用 AT 解码器输出进行知识蒸馏。其他策略是在第 III-B 节中介绍的策略。粗体数字表示最佳词错误率结果。

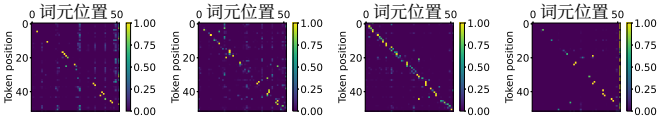
无语言模型的模型	ConvEnc.	InterCTC	ConvDec.	InterCE	Mask Exp.	KD	dev-干净	dev-其他	测试集-干净	测试集-其他	*WERR*
AT							3.4	8.1	3.6	8.0	-
	✓						2.7	6.9	2.9	6.9	13.8%
	✓	✓					2.7	6.8	2.8	6.7	16.3%
CASS-NAT							3.6	8.8	3.8	8.6	-
	✓						3.0	7.3	3.2	7.5	12.8%
	✓	✓					2.9	7.4	3.1	7.3	15.1%
			✓				3.4	8.3	3.6	8.4	2.3%
			✓	✓			3.3	8.2	3.6	8.3	3.5%
	✓	✓	✓	✓			2.8	7.1	2.9	7.1	17.4%
					✓		3.6	8.9	3.8	8.7	-1.2%
			✓	✓	✓		3.3	8.3	3.6	8.1	5.8%
	✓	✓	✓	✓	✓		**2.7**	**7.1**	**2.9**	**7.0**	18.6%
						✓	3.6	8.8	3.8	8.5	1.2%

以及 AT 基线，但没有观察到任何改进，因此未纳入其结果。当将 ConvEnc. 和 InterCTC 应用于 CASS-NAT 时，观察到与将它们用于自回归模型时相似的改进。这些改进源于它们能够捕捉声学表示中的局部细节，因此能够学习到更强的编码器，并为解码器中的隐式词元级声学嵌入建模提供准确的 CTC 对齐。

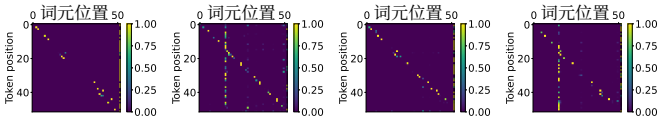
其次，使用 ConvDec. 和 InterCE 后，CASS-NAT 在 test-other 数据集上的词错误率进一步降至 7.1%，相较于 CASS-NAT 基线，词错误率相对改善了 17.4%。然而，ConvDec. 和 InterCE 的效果不如 ConvEnc. 和 InterCTC，这表明在 CASS-NAT 中，对帧级声学表示进行更强的建模，比对词元级声学表示（TAE）进行建模更为重要。这是合理的，因为 TAE 是从编码器的帧级声学表示中提取的。有趣的是，我们观察到单独应用触发掩码扩展方法时，词错误率相对下降了 -1.2%。然而，该方法与 ConvDec. 相关策略结合使用时，能够改善词错误率。原因可能是，如果上下文建模不够强，对每个词元扩展声学边界可能会使解码器产生混淆。

此外，如表 IV 最后一行所示，采用知识蒸馏（以自回归模型的输出作为非自回归训练的目标）并未改善词错误率。尽管知识蒸馏已被证明对非自回归神经机器翻译有益，但在 [53], 中，它似乎并不适用于语音识别，可能是因为对于同一个输入并不存在不同的输出，而在机器翻译中，源语言在目标语言中存在多种译法。

最后，采用所有训练策略后，与 CASS-NAT 基线相比，我们的词错误率相对降低了 18%；与表现最佳的 AT基线相比，词错误率仅高出 3%。总之，我们得到了一种性能接近其具有显著 GPU加速比的自回归对应模型。



（a）最后一个 SAD 中的自注意力 $5^{th} \sim 8^{th}$ 头



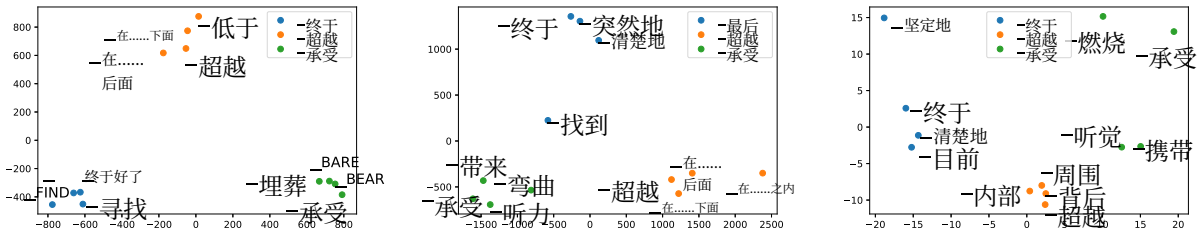
（b）最后一个 MAD 中的自注意力 $5^{th} \sim 8^{th}$ 头

图 5：最后一个自注意力解码器（SAD）和混合注意力解码器（MAD）中第 $5^{th} - 8^{th}$ 个注意力头的注意力权重分布。使用了 LiriSpeech 训练集 clean-100 子集中的第一条话语。纵轴表示输出词元

E. CASS-NAT 为何有效？——解码器分析

本节将通过分析 CASS-NAT 解码器中的以下要素，解释 CASS-NAT 为何能够取得接近其自回归对应模型的良好性能：(1) 注意力权重分布；以及 (2) 声学层面的声学表示。

首先，为了分析 CASS-NAT 解码器中注意力机制的行为，我们选取 LibriSpeech train-clean-100 子集中的第一个话语，并绘制最后一个 SAD 和 MAD 模块中的注意力权重分布。图 5 展示了最后 4 个注意力头（总共 8 个注意力头）的权重。对于 SAD 中的自注意力权重分布，我们从图中发现，大多数注意力头都学到了词元级声学表示之间的单调对齐，这表明每个词元都依赖于相邻词元。这与使用连续词袋（CBOW）和跳字模型 [54] 进行词嵌入的思路相似。单调对齐也表明了相对位置编码的有效性，因为语义相似度较高但距离较远的词元



(a) TAEE 输出 (b) SAD 输出 (c) MAD 输出

图 6：分别对 TAEE、SAD 和 MAD 的输出中三个子词单元的词元级声学嵌入进行可视化。使用主成分分析（PCA）绘制这些嵌入。

具有较低的注意力权重。在 MAD 的注意力权重方面，注意力计算的若干子空间中呈现出不同的行为，输出可能依赖于同一个输入（图 5b 中的竖线）。

如第二节-B2 所述，我们假设 CASS-NAT 解码器能够利用词元级声学嵌入捕捉语言语义。为验证这一假设，我们针对每个词元（本例中为子词单元），分别从 TAEE、SAD 和 MAD 计算三种不同的词元级声学嵌入。具体而言，我们首先从 train-clean-100 Librspeech 子集中提取前 5000 个话语的词元级声学表征，每个表征都带有一个子词单元形式的标签。每个子词单元的嵌入是相应表征的平均值。随后，我们随机选择三个子词单元，并为每个子词单元找出与其最接近的四个子词单元（以余弦相似度距离衡量）。借鉴词嵌入的可视化方法，将选出的 12 个嵌入降至二维空间（使用主成分分析（PCA）），然后绘制出来。图 6 展示了 CASS-NAT 解码器不同层级的嵌入示例。图中表明，词元级声学嵌入不仅能学习声学相似性（**BEAR** 与**BARE**），还能学习词元级语义相似性（**BENEATH** 与**BELOW**）。尽管解码器不同层级的嵌入提供的信息各不相同，仍会出现这种情况。例如，较高层（MAD）的嵌入更关注语法相似性（如**FIRMLY**与**FINALLY**），这与词嵌入类似。这一发现表明，可以在共同空间中学习联合的语音与文本嵌入。尽管没有显式的语言建模，CASS-NAT 解码器仍能够学习有意义的嵌入，这或许可以解释为何其性能与对应的自回归模型相近。

F. 其他数据集上的结果

最后，我们将所提出的 CASS-NAT 应用于另外三个数据集：Aishell（普通话语音）、TED2（英语自发语音）和 MyST（儿童语音），以展示 CASS-NAT 的泛化能力。此外，我们还探究了 CASS-NAT 编码器采用不同初始化方案的影响，并测量了 CASS-NAT 相较于 AT基线的加速比。所有结果汇总于表 V。如表中所示，在 Aishell1、TED2 和 MyST 数据集上，采用随机初始化编码器的 CASS-NAT 性能最佳；而在 LibriSpeech 数据集上，采用预训练编码器的表现更好。原因可能是，出于时间方面的考虑以及确保公平比较，我们将 CASS-NAT 的训练迭代次数设为与 AT基线相同；然而，与其他数据集相比，LibriSpeech 是一个规模较大的数据库，因此需要更多训练迭代，才能为解码器获得准确的对齐。因此，我们建议在大型数据集上训练 CASS-NAT 时使用预训练编码器。

采用适当的编码器初始化后，我们将 CASS-NAT 的结果与 AT基线以及采用非自回归模型的最先进水平结果进行比较。首先，与采用贪心搜索解码的 AT基线相比，CASS-NAT 的词错误率表现更好（在 Aishell1、TED2 和 MyST 上）或相近（在 LibriSpeech 上），实时因子加速比为 2.89 倍。与采用束搜索解码的 AT基线相比，CASS-NAT 在 Aisehll 和 TED2 上的词错误率表现仍然更好，而在 LibriSpeech 和 MyST 上的词错误率则接近基线，但实时因子加速比达到 24 倍。其次，与文献中的其他非自回归方法相比，从表中可以看出，在未使用预训练声学模型（例如 Wav2vec2.0 [37]）或语言模型（例如 BERT [59]）的情况下，本文提出的 CASS-NAT 在 LibriSpeech、TED2 和 MyST 上取得了新的最先进结果。为与使用额外训练数据（例如使用未标注数据进行半监督学习）的结果进行比较，我们将自监督学习预训练模型用作 CASS-NAT 的编码器，以检验其能否超越表 V 中的最先进水平结果。受时间和计算资源限制，我们仅获得了 Aishell1 和 MyST 数据集上的结果。结果表明，采用自监督学习预训练的 CASS-NAT 可取得与文献中最佳非自回归模型相当的结果 [57], [58] 。

VI. 结论

本文全面研究了 CASS-NAT 框架。该框架在 [23], 中提出，利用从 CTC 输出空间获得的对齐来提取词元级声学嵌入（TAE），并将 TAE 视为自回归 Transformer（AT）中词嵌入的替代，从而实现并行计算。在训练过程中，使用维特比对齐估计后验概率，

表 V：CASS-NAT 使用不同编码器作为起点时的词错误率（WER）结果，包括随机初始化（随机编码器）、使用 CTC 训练的编码器初始化（CTC 编码器），以及使用 AT 编码器初始化（AT 编码器）。表中还列出了采用贪心解码和束搜索解码的 AT 基线，以及文献中使用非自回归方法取得的最先进水平结果。“-”表示文献中未报告词错误率结果。实时因子（RTF）在图形处理器上以批大小为 1 进行评估。粗体数字表示 CASS-NAT 和未使用自监督预训练的最先进水平方法所取得的最佳结果。

模型	条件	推理速度		LibriSpeech				Aishell1		TED2		MyST	
		RTF	速度	dev		test		dev	test	dev	test	dev	test
				干净	其他	干净	其他						
AT	贪心搜索	0.0387	1.00x	2.7	7.0	2.9	7.0	5.1	5.9	9.8	7.8	18.1	17.0
	束搜索	0.3348	0.12x	2.7	6.8	2.8	6.7	4.7	5.1	8.2	7.6	16.5	16.2
先前非自回归结果	不使用半监督学习			2.8	7.3	3.1	7.2 [24]	4.5	4.9 [17]	8.7	8.0 [55]	28.0	27.8 [56]
	使用半监督学习			1.7	3.6	1.8	3.6 [22]	4.1	4.5 [57]	-	-	16.8	16.5 [58]
CASS-NAT（本文工作）	随机编码器	0.0134	2.89倍	3.3	8.4	3.6	8.3	**4.7**	**5.0**	8.4	**7.5**	**16.8**	**16.4**
	CTC编码器			2.8	7.2	3.1	7.3	5.2	5.6	**8.0**	7.7	17.1	16.7
	AT编码器			**2.7**	**7.1**	**2.9**	**7.0**	5.1	5.4	8.2	7.8	17.1	16.7
	HuBERT编码器			-	-	-	-	4.2	4.5	-	-	16.0	15.6

并采用多种训练策略来提升 CASS-NAT 的词错误率（WER）表现。在推理过程中，我们深入研究了近期在 [23] 中提出的基于误差的采样对齐（ESA）方法，以减少训练与推理之间的对齐不匹配。详细实验表明：（1）混合注意力解码器（MAD）对于降低 WER 至关重要；（2）ESA 解码效果良好的原因在于其长度预测错误率更低；（3）卷积增强编码器和解码器、中间层损失以及掩码扩展能够改善 CASS-NAT 的 WER，而知识蒸馏则不能；（4）TAE 与词嵌入具有相似的功能，例如表示语法结构，这表明无需语言模型也有可能学习语义。因此，在若干自动语音识别任务中，无论是否使用自监督学习预训练，CASS-NAT 在非自回归模型中均取得了新的最先进结果。数据集涵盖英语和普通话语音、朗读语音和自发语音，以及儿童和成人语音，展现了所提方法的泛化能力。相较而言，CASS-NAT 的表现与采用束搜索解码的自回归模型相当，同时保持 ∼24倍加速。未来的工作将研究如何利用外部数据引入语言模型。

参考文献

[1] J. Li, Y. Wu, Y. Gaur, C. Wang, R. Zhao, 和 S. Liu，“大规模语音识别中流行端到端模型的比较”，*Interspeech 2020*，第 1–5 页，2020年。[2] J. Li 等，“端到端自动语音识别的最新进展”，*APSIPA*信号与信息处理汇刊，第 11 卷，第 1 期，2021年。[3] A. Graves、S. Fern´andez、F. Gomez 和 J. Schmidhuber，“连接时序分类：利用循环神经网络标注未分段的序列数据”，载于第*23*届国际机器学习会议（*ICML*）论文集，2006年，第 369–376 页。[4] W. Chan、N. Jaitly、Q. Le 和 O. Vinyals，“听、关注与拼写：一种用于大词汇量会话语音识别的神经网络”，载于*IEEE*国际声学、语音与信号处理会议（*ICASSP*）。IEEE，2016年，第 4960–4964 页。[5] A. Graves，“循环神经网络的序列转导”，国际机器学习会议（*ICML*），2012年。[6] Q. Zhang、H. Lu、H. Sak、A. Tripathi、E. McDermott、S. Koo 和 S. Kumar，“Transformer转导器：一种采用Transformer编码器和RNN-T损失的流式语音识别模型”，载于*IEEE*国际声学、语音与信号处理会议（*ICASSP*）。IEEE，2020年，第 7829–7833 页。

[7] A. Vaswani、N. Shazeer、N. Parmar、J. Uszkoreit、L. Jones、A. N. Gomez、Ł. Kaiser 和 I. Polosukhin，《Attention Is All You Need》，载于第*31*届神经信息处理系统国际会议论文集（*NIPS*），2017年，第 6000–6010 页。[8] L. Dong、S. Xu 和 B. Xu，《Speech-transformer：一种用于语音识别的无循环序列到序列模型》，载于*IEEE*国际声学、语音与信号处理会议（*ICASSP*）。IEEE，2018年，第 5884–5888 页。[9] 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（*ICLR*），2018年。[10] J. Lee、E. Mansimov 和 K. Cho，《通过迭代细化实现确定性非自回归神经序列建模》，载于自然语言处理经验方法会议（*EMNLP*），2020年，第 1173–1182 页。[11] C. Saharia、W. Chan、S. Saxena 和 M. Norouzi，《利用潜在对齐进行非自回归机器翻译》，载于自然语言处理经验方法会议（*EMNLP*），2020年，第 1098–1108 页。[12] W. Qi、Y. Gong、J. Jiao、Y. Yan、W. Chen、D. Liu、K. Tang、H. Li、J. Chen、R. Zhang等，《Bang：通过大规模预训练衔接自回归与非自回归生成》，载于国际机器学习会议（*ICML*）。PMLR，2021年，第 8630–8639 页。[13] N. Chen、S. Watanabe、J. A. Villalba、P. Zelasko 和 N. Dehak，《用于语音识别的非自回归 Transformer》，*IEEE*信号处理快报（*SPL*），2020年。[14] 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》，*Interspeech 2020*论文集，第 3655–3659 页，2020年。[15] Y. Bai、J. Yi、J. Tao、Z. Tian、Z. Wen 和 S. Zhang，《通过非自回归模型和从 BERT 进行跨模态知识迁移，实现快速端到端语音识别》，*IEEE/ACM*音频、语音与语言处理汇刊（*TASLP*），第29卷，1911，2021年，第 1897 页。[16] Z. Tian、J. Yi、J. Tao、Y. Bai、S. Zhang 和 Z. Wen，《用于端到端语音识别的脉冲触发非自回归 Transformer》，*Interspeech 2020*论文集，第 5026–5030 页，2020年。[17] F. Yu、H. Luo、P. Guo、Y. Liang、Z. Yao、L. Xie、Y. Gao、L. Hou 和 S. Zhang，《面向基于 CIF 的非自回归端到端自动语音识别的边界与上下文感知训练》，载于自动语音识别与理解研讨会（*ASRU*）。IEEE，2021年，第 328–334 页。[18] N. Chen、P. ˙Zelasko、L. Moro-Vel´azquez、J. Villalba 和 N. Dehak，《Align-Denoise：单遍非自回归语音识别》，载于*Interspeech 2021*论文集，2021年，第 3770–3774 页。[19] E. A. Chi、J. Salazar 和 K. Kirchhoff，《Align-refine：通过迭代重新对齐实现非自回归语音识别》，*2021*年计算语言学协会北美分会会议：人类语言技术论文集（*NAACL-HLT*），第 1920–1927 页，2020年。[20] W. Chan、C. Saharia、G. Hinton、M. Norouzi 和 N. Jaitly，《Imputer：通过插补和动态规划进行序列建模》，载于国际机器学习会议（*ICML*）。PMLR，2020年，第 1403–1413 页。[21] J. Nozaki 和 T. Komatsu，《放宽基于中间预测进行条件化的基于 CTC 的自动语音识别中的条件独立性

假设》，载于 Interspeech 2021论文集，2021年，第 3735–3739 页。[22] E. G. Ng、C.-C. Chiu、Y. Zhang 和 W. Chan，《突破非自回归语音识别的极限》，载于 Interspeech论文集，第 3725–3729 页，2021年。[23] R. Fan、W. Chu、P. Chang 和 J. Xiao，《Cass-nat：基于 CTC 对齐的单步非自回归语音识别 Transformer》，载于 IEEE国际声学、语音与信号处理会议（ICASSP）。IEEE，2021年，第 5889–5893 页。[24] R. Fan、W. Chu、P. Chang、J. Xiao 和 A. Alwan，《改进的单步非自回归自动语音识别 Transformer》，载于 Interspeech 2021论文集，2021年，第 3715–3719 页。[25] W.-N. Hsu、Y.-H. H. Tsai、B. Bolte、R. Salakhutdinov 和 A. Mohamed，《HuBERT：糟糕的教师能为自动语音识别预训练带来多大帮助？》，载于 IEEE国际声学、语音与信号处理会议（ICASSP）。IEEE，2021年，第 6533–6537 页。[26] T. Nakatani，《通过连接主义时序分类和语言模型集成改进基于 Transformer 的端到端语音识别》，载于 Interspeech论文集，2019年，第 1408–1412 页。[27] P. Zhou、R. Fan、W. Chen 和 J. Jia，《通过并行调度采样和相对位置嵌入提升 Transformer 语音识别的泛化能力》，arXiv 预印本 arXiv:1911.00203，2019年。[28] I. Bello、B. Zoph、A. Vaswani、J. Shlens 和 Q. V. Le，《注意力增强卷积网络》，载于 IEEE/CVF 国际计算机视觉会议（ICCV）论文集，2019年，第 3286–3295 页。[29] A. W. Yu、D. Dohan、M.-T. Luong、R. Zhao、K. Chen、M. Norouzi 和 Q. V. Le，《QANet：结合局部卷积与全局自注意力的阅读理解》，载于国际学习表征会议（ICLR），2018年。[30] 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 Transformer》，Interspeech 2020论文集，第 5036–5040 页，2020年。[31] W. Han、Z. Zhang、Y. Zhang、J. Yu、C.-C. Chiu、J. Qin、A. Gulati、R. Pang 和 Y. Wu，《ContextNet：利用全局上下文改进用于自动语音识别的卷积神经网络》，Interspeech 2020 论文集，第 3610–3614 页，2020年。[32] C. Szegedy、W. Liu、Y. Jia、P. Sermanet、S. Reed、D. Anguelov、D. Erhan、V. Vanhoucke 和 A. Rabinovich，《深入探索卷积》，载于 IEEE 计算机视觉与模式识别会议（CVPR）论文集，2015年，第 1–9 页。[33] A. Tjandra、C. Liu、F. Zhang、X. Zhang、Y. Wang、G. Synnaeve、S. Nakamura 和 G. Zweig，《Deja-vu：深度 Transformer 网络中的双重特征呈现与迭代损失》，载于 IEEE国际声学、语音与信号处理会议（ICASSP）。IEEE，2020年，第 6899–6903 页。[34] J. Lee 和 S. Watanabe，《基于 CTC 的语音识别的中间损失正则化》，载于 IEEE国际声学、语音与信号处理会议（ICASSP）。IEEE，2021年，第 6224–6228 页。[35] J. Lee、J. Kang 和 S. Watanabe，《通过中间 CTC 按需进行层剪枝》，载于 Interspeech 2021论文集，2021年，第 3745–3749 页。[36] Y. Wang、A. Mohamed、D. Le、C. Liu、A. Xiao、J. Mahadeokar、H. Huang、A. Tjandra、X. Zhang、F. Zhang 等，《用于混合语音识别的基于 Transformer 的声学建模》，载于 IEEE国际声学、语音与信号处理会议（ICASSP）。IEEE，2020年，第 6874–6878 页。[37] A. Baevski、H. Zhou、A. Mohamed 和 M. Auli，《wav2vec 2.0：语音表征自监督学习框架》，NeurIPS 2020，2020年。[38] S. Watanabe、T. Hori、S. Kim、J. R. Hershey 和 T. Hayashi，《用于端到端语音识别的混合 CTC/注意力架构》，IEEE信号处理专题精选期刊（JSTSP），第 11 卷，第 8 期，第 1240–1253 页，2017年。[39] N. Moritz、T. Hori 和 J. Le，《使用 Transformer 模型进行流式自动语音识别》，载于 IEEE国际声学、语音与信号处理会议（ICASSP）。IEEE，2020年，第 6074–6078 页。[40] A. Zeyer、A. Merboldt、R. Schl¨uter 和 H. Ney，《改进神经转导器的新训练流程》，Interspeech 2020论文集，第 2812–2816 页，2020年。[41] B. Yang、L. Wang、D. F. Wong、L. S. Chao 和 Z. Tu，《卷积自注意力网络》，载于《2019年计算语言学协会北美分会会议：人类语言技术论文集》第1卷（长文与短文）

，2019年6月，第4040页–4045。

[42] Z. Dai、Z. Yang、Y. Yang、J. G. Carbonell、Q. Le 和 R. Salakhutdi- nov，“Transformer-xl：超越固定长度上下文的注意力语言模型”，载于计算语言学协会第57届年会（ACL）论文集，2019年，第2978–2988页。[43] V. Panayotov、G. Chen、D. Povey 和 S. Khudanpur，“Librispeech：基于公有领域有声书的自动语音识别语料库”，载于 IEEE 国际声学、语音与信号处理会议（ICASSP）。IEEE，2015年，第5206–5210页。[44] H. Bu、J. Du、X. Na、B. Wu 和 H. Zheng，“Aishell-1：开放源代码普通话语音语料库与语音识别基线”，载于国际语音数据库及语音输入/输出系统与评估协调委员会东方分会第20届会议（O-COCOSDA）。IEEE，2017年，第1–5页。[45] A. Rousseau、P. Del´eglise、Y. Esteve 等，“通过精选数据增强 TED-LIUM 语料库，用于语言建模并收录更多 TED演讲。”载于第九届国际语言资源与评估会议（LREC）论文集，2014年，第3935–3939页。[46] W. Ward、R. Cole、D. Bolanos、C. Buchenroth-Martin、E. Svirsky、S. V. Vuuren、T. Weston、J. Zheng 和 L. Becker，“My Science Tutor：一款面向小学科学的对话式多媒体虚拟辅导教师”，《ACM语音与语言处理汇刊》（TSLP），第7卷，第4期，第1–29页，2011年。[47] T. Kudo 和 J. Richardson，“Sentencepiece：一种用于神经文本处理的简便、语言无关的子词分词器和去分词器”，《自然语言处理经验方法会议》（EMNLP），第66页，2018年。[48] T. Ko、V. Peddinti、D. Povey 和 S. Khudanpur，“用于语音识别的音频增强”，载于国际语音通信协会第十六届年会，2015年，第3586–3589页。[49] D. S. Park、W. Chan、Y. Zhang、C.-C. Chiu、B. Zoph、E. D. Cubuk 和 Q. V. Le，“Specaugment：一种用于自动语音识别的简易数据增强方法”，Interspeech 2019论文集，2617，2019年，第2613页–。[50] R. Fan、P. Zhou、W. Chen、J. Jia 和 G. Liu，“一种用于语音识别的在线基于注意力的模型”，Interspeech 2019论文集，4394，2019年，第4390页–。[51] A. Paszke、S. Gross、F. Massa、A. Lerer、J. Bradbury、G. Chanan、T. Killeen、Z. Lin、N. Gimelshein、L. Antiga 等，“Pytorch：一种命令式风格、高性能的深度学习库”，载于神经信息处理系统国际会议（NIPS），2019年，第8026–8037页。[52] B. Zhang、H. Lv、P. Guo、Q. Shao、C. Yang、L. Xie、X. Xu、H. Bu、X. Chen、C. Zeng 等，“Wenetspeech：用于语音识别的 10000+ 小时多领域普通话语料库”，载于 IEEE 国际声学、语音与信号处理会议（ICASSP）。IEEE，第6182–6186页。[53] C. Zhou、J. Gu 和 G. Neubig，“理解非自回归机器翻译中的知识蒸馏”，载于国际学习表征会议（ICLR），2020年。[54] T. Mikolov、K. Chen、G. Corrado 和 J. Dean，“在向量空间中高效估计词表示”，国际学习表征会议（ICLR），2013年。[55] Y. Higuchi、N. Chen、Y. Fujita、H. Inaguma、T. Komatsu、J. Lee、J. Nozaki、T. Wang 和 S. Watanabe，“语音到文本生成中非自回归建模的比较研究”，自动语音识别与理解（ASRU），第47–54页，2021年。[56] R. Fan、Y. Zhu、J. Wang 和 A. Alwan，“迈向更好的自监督模型领域适应：儿童自动语音识别案例研究”，《IEEE 信号处理专题期刊》，第16卷，第6期，第1242–1252页，2022年。[57] K. Deng、Z. Yang、S. Watanabe、Y. Higuchi、G. Cheng 和 P. Zhang，“利用预训练声学模型和语言模型改进非自回归端到端语音识别”，IEEE 国际声学、语音与信号处理会议（ICASSP），第8522–8526页，2022年。[58] R. Fan 和 A. Alwan，“DRAFT：一种减少自监督学习中领域偏移的新型框架及其在儿童自动语音识别中的应用”，载于 Interspeech 2022论文集，2022年，第4900–4904页。[59] J. Devlin、M.-W. Chang、K. Lee 和 K. Toutanova，“Bert：用于语言理解的深度双向Transformer预训练”，计算语言学协会北美分会年会（NAACL-HLT），第4171–4186页，2019年。