一种更好更快的流式ASR端到端模型

*Bo Li\*、Anmol Gulati\*、Jiahui Yu*[[1]](" \l "_ftn1" \o ")*，Tara N.Sainath，Chung Cheng Chiu，Arun Narayanan，*

*张硕、彭若明、何燕章、秦琰、韩炜、乔亮、张宇、斯特罗曼、吴永辉*

谷歌有限责任公司，美国

{波波里、安莫古拉提、嘉惠峪}@谷歌网站

# 摘要

端到端（E2E）模型在许多方面都优于流式语音识别的现有传统模型[1]，包括质量（通过字错误率（WER）测量）和端点延迟[2]。然而，与传统的ASR模型相比，该模型仍然倾向于将预测延迟到最后，因此具有更高的部分延迟。为了解决这个问题，我们考虑通过一个名为FastEmit的算法来鼓励E2E模型尽早发出单词[3]。当然，延迟的改善会导致质量下降。为了解决这个问题，我们探索用一致性层替换E2E模型编码器中的LSTM层[4]，这显示了ASR的良好改进。其次，我们还探讨了运行第二遍波束搜索以提高质量。为了确保第二遍快速完成，我们探索了非因果一致性层，这些一致性层馈送到相同的1stpass RNN-T解码器中，这是一种称为级联编码器的算法[5]。总的来说，Conformer RNN-T和级联编码器为流式ASR提供了更好的质量和延迟权衡。

*索引扩展*-RNN-T，一致性，级联编码器，延迟

# 1.    导言

拥有一个端到端（E2E）语音识别模型[6–11]，在质量和延迟的所有维度上与传统模型（具有独立的声学、发音和语言模型组件）的质量相匹配/超过，已经成为过去几年一个具有挑战性的研究方向。

在延迟方面，E2E模型的一个这样的要求是它们必须是流式的。当用户说话时，它们必须以最小的延迟向屏幕发出单词。研究了递归神经网络变换器（RNN-T）[9]、神经变换器[12,13]、单调注意[14,15]、触发注意[16]和Scout网络[17]等模型来解决流的性质。另一个要求是，一旦用户停止说话，模型必须快速停止解码，这就是所谓的端点问题。将endpointer折叠到E2E模型中有助于解决这个问题[2]。

在质量方面，我们需要一种比传统的模型能获得更好的字错误率（WER）的模型。使用第一遍流式RNN-T模型和第二遍非流式注意力模型有助于解决质量问题[18]。此外，通过上下文偏向提高了模型识别专有名词的性能[19]。

将上述许多组件放在一起，我们最近发布了一个在设备上的模型，该模型在WER和端点延迟方面都超过了传统模型[1]。然而，该模型的一个问题是，与通过校准训练的传统模型相比，部分假设仍然延迟到最后。

在这项工作中，我们探索的想法，以改善模型的部分延迟。语音-文本对齐可用于改进令牌预测定时[20,21]，这通常会导致较大的质量下降。我们采用了最近提出的FastEmit[3]。它正则化RNN-T格子来鼓励非空白令牌，并基于转发概率来跨越整个序列的空白令牌。结果表明，该方法能有效地减少模型的部分延迟，且质量下降较少。

可以预期，任何试图改善部分延迟的研究都会导致轻微的质量下降。为了解决这个问题，我们探索改进核心E2E模型，将编码器中的循环LSTM层替换为一致性层，这在Librispeech[4]中显示出了希望。此外，我们还考虑改进我们的第二遍策略，以便可以运行波束搜索，而不是重新扫描[5]。我们在语音搜索任务中探索了改进延迟和质量的方法，与[1]相比，平均部分延迟和预取延迟分别减少了250ms和170ms，相对功耗降低了10%。

# 2.    模型改进

## 2.1.    符合者

我们使用Conformer[4]来提高模型的质量。文[4]中使用的模型是针对非流、单域和短句的任务设计的。在这项工作中，我们将其用于流式多域任务[1]。我们首先将自我注意、卷积和规范化层限制为仅在流应用程序的先前上下文中使用。我们还用局部自我注意取代了全语境自我注意，这有助于推广到长话语[22]。与[1]一样，我们对来自不同领域的数据组合进行系统培训。为了提高效率，我们在创建批处理（即bucketing）之前聚合长度相似的语句。由于跨域的长度分布不同，对于特定批次，数据可能会偏向某个域。这可能导致在原始一致性层中使用的批处理规范化的统计有偏差，进而限制模型跨域的泛化能力。我们通过将批处理规范化层替换为组规范化层来解决这个问题[23]。这对于我们的多域模型的质量至关重要。

此外，我们通过去除原始的相对位置编码来加速训练和推理[24]。相反，我们重用现有的卷积模块，该模块从相邻上下文聚合信息，从而隐式地提供相对位置信息[24，25]。这是通过简单地交换卷积模块和多头自我注意模块的顺序来实现的，从而得到图1所示的改进的一致性块。



图1：改进的一致性块体系结构。

## 2.2.    双通道级联编码器

双道次模型的一个大目标是弥补第一道次的质量下降。为了改进我们目前的两遍方法，也就是运行重扫描，我们希望运行波束搜索。这种波束搜索的挑战之一是，我们需要该模型对短搜索语句以及长字幕语句的解码具有鲁棒性。通常，基于注意力的模型在长形式上做得很差[22]。

为了解决这个问题，我们考虑使用RNN-T本身运行第二次通过波束搜索，它对长格式具有更好的鲁棒性[26]。我们称之为级联编码器[5]的模型如图2所示。为了将LAS的非因果方面（提高质量）[18]添加到流式因果RNN-T系统中，我们在因果编码器层的顶部添加了额外的非因果编码器层。第1遍仅使用因果编码器和RNN-T解码器。在第二遍中，附加的非因果层接收第一遍编码器输出的左右上下文，并再次馈送到同一解码器。一个RNN-T解码器在第一和第二通道之间共享，以实现更小的模型尺寸和设备上的优势。

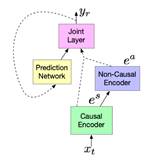


图2：级联编码器模型的框图。

# 3.    延迟改善

在本节中，我们将介绍流式ASR的延迟度量以及减少延迟的技术。与平均令牌延迟不同[21]，我们采用了与流式语音应用程序（如语音搜索和助手）直接相关的度量。

## 3.1.    终结点延迟

在基于ASR的流应用程序中，当用户结束讲话时是触发下一步实现的关键信号。对于语音搜索/助理，它向ASR系统发送信号，以最终确定识别结果，并启动服务器请求更多信息或客户端的操作。Endpointer latency度量用户结束讲话和系统预测查询结束（EOQ）之间的时间差。这是我们在前面的工作[1,2]中通常提到的延迟度量。较低的终结点延迟导致较低的应用程序延迟和更快的系统响应。我们采用相同的RNN-T端点模型，并在训练期间对EOQ进行早期和晚期惩罚，以减少端点延迟[2]。我们报告了中位数（即第50百分位，EP50）和第90百分位（EP90）的潜伏期进行比较。

## 3.2.    预取延迟

预取是一种常用的技术，用于基于web的应用程序中，以减少应用程序堆栈中涉及的网络延迟[27,28]。流应用程序通常不等待EOQ信号，而是基于部分/不完全识别假设发送早期服务器请求（检索相关信息或下一步操作所需的资源）。如果预取基于与生成EOQ时的最终假设不匹配的部分假设，那么它将作为过早预取而被丢弃，从而增加总体计算成本。但是，如果预取中使用的部分假设与最终结果匹配，则系统可以立即执行以下操作，而无需等待其他服务器响应。这节省了从服务器计算和获取响应的时间，并减少了网络处理延迟。预取延迟定义为触发第一次正确预取和用户结束讲话之间的时间差[29]。只有当预取使用的部分识别结果与最终识别结果匹配时，预取才是正确的。如果找不到正确的预取，则延迟将回落到Endpointer延迟。除了延迟，预取率（即每个查询的预取次数）也是一个重要的度量。为了进行比较，我们报告了第50和第90百分位延迟（PF50和PF90）以及预取率（PFR）。

### 3.2.1.    E2E预取

触发预取的方法有多种[29]。它通常依赖于从解码或外部语音活动检测器（VAD）观察固定的静默间隔。最近有人提出了一种E2E预取方法，它产生了更好的预取延迟和预取速率折衷[29]。它利用RNN-T端点模型的EOQ预测来进行预取决策。与端点不同，我们希望为每个部分结果生成EOQ，但避免改变解码。为了实现这一点，在顶梁上显示EOQ之前的每一帧，我们人为地将其添加到当前假设的末尾，以计算给定观测值和迄今为止最佳路径的概率。这使我们能够预测查询在每个时间帧完成的可能性，以及在解码后续帧时相应的部分结果。如果概率大于预定义的阈值，则系统声明预取决策。

## 3.3.    部分延迟

预取依赖于ASR模型生成的部分结果。最好是输出延迟较小的部分假设的系统。因此，我们将部分延迟定义为模型生成第一个正确的部分假设与用户结束讲话之间的时间差。同样，部分假设的正确性是指它是否符合最终确认的假设。与预取延迟不同，部分延迟是ASR模型本身固有的，不依赖于应用程序管道中实现的附加逻辑，例如触发优化的决策。部分延迟是预取延迟的下限。开发减少模型部分延迟的技术将为减少整体应用程序延迟提供更大的空间，并且独立于实际的应用程序管道。

### 3.3.1.    约束对齐

改善部分延迟的一种方法是基于从其他模型生成的对齐来约束模型的预测[20]。与约束EOU令牌的RNN-tep模型类似，如果每个单词的开始和结束令牌从围绕从对齐中获得的相应基本真值的缓冲区中掉出来，则约束对齐会惩罚它们的预测。它已经被证明可以产生更准确的预测定时信息，并且可以用来改善部分延迟。

### 3.3.2.    快速发射

在传统的RNN-T优化中，只要目标序列的日志概率最大化，则发射非空白令牌和空白令牌相等地对待。然而，在流式ASR系统中，应该阻止空白令牌（即，输出没有），因为它会导致更高的延迟。因此发展了一种简单有效的序列级发射正则化技术FastEmit[3]。它鼓励预测非空白标记，并基于前向和后向概率在整个序列中抑制空白标记。通过在[30]中的原始方程20中添加正则化超参数来实现（详细概念定义请参见[30]）：*λ*快速发射



*∂*Lα（t，u）(1+ λ快速发射)*β*（t，u+1）如果*k公司*=y*u型*+1

*∂*公共关系（k | t，u）=-公共关系（yx |）β（t+1，u）如果*k公司*=∅

否则为0。

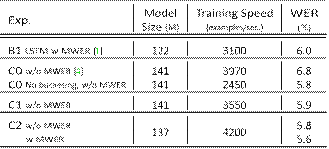
直观地说，当反向传播到流式ASR网络时，FastEmit将“更高的学习率”（>0）应用于非空白令牌的预测，而空白令牌的预测保持不变。在不同的RNN-T模型上，它能显著降低延迟，同时保持识别准确率。更重要的是，FastEmit不需要任何事先的对齐信息[20,21]，也没有额外的培训或服务成本。*λ*快速发射

# 4.    实验装置

模型架构：所有模型都使用128D logMel特征前端进行训练[26]。这些特征是使用32ms窗口和10ms跳数计算的。来自4个连续帧的特征被叠加以形成512D输入表示，该输入表示进一步由因子3进行二次采样。然后附加一个16D一个热域id向量以形成528D输入向量作为模型的输入。在[4]之后，我们在编码器中使用512D一致性层。一致性层采用因果卷积和左上下文注意层，严格限制模型不使用未来输入。自我注意层采用8头注意，卷积核大小为15。编码器由17个一致层组成。RNN-T解码器由一个具有2个LSTM层的预测网络（2048个单元向下投影到640个输出单元）和一个具有单个前馈层（640个单元）的联合网络组成。在解码器之前的流式编码器顶部的级联编码器中使用了两个非因果一致性层，它们接受额外的5.04s右上下文。所有的E2E模型都经过训练，可以预测4096个单词[31]。解码用第二遍MaxEnt语言模型重新扫描[32]。

一致性RNN-T模型有137M个参数。级联编码器模型中的两个非因果一致性层引入了额外的12M参数。所有模型都在Tensorflow[33]中使用Lingvo[34]工具箱在8×8张量处理单元（TPU）切片上进行训练，全局批量大小为4096个语句。使用同步随机梯度下降和Adam优化器优化模型[35]。

数据集：所有的E2E模型都是在与[1]相同的音频-文本对上训练的，与传统LM训练的万亿字纯文本数据相比，这只是数据的一小部分。为了增加词汇和训练数据的多样性，我们使用表1：不同编码器的RNN-T模型的性能进行了探讨。



如[26]所述，通过合并多域话语获得更多数据。这些多领域的话语跨越了搜索、farfield、电话和YouTube等领域。所有的数据集都是匿名的，而且都是手工转录的；YouTube话语的转录是以半监督的方式进行的[36]。除了不同的训练集，多条件训练（MTR）[37]和随机数据下采样到8kHz[38]也被用来进一步增加数据的多样性。噪声数据产生的信噪比为0～30db，平均信噪比为12db，T60次为0～900ms，平均500ms。生成8khz和16khz两个版本的数据，每个都具有相同的概率，以使模型对不同的采样率具有鲁棒性。

测试集包括约12K个语音搜索语句，持续时间小于5.5s。它们是匿名的，并且是手写的，是谷歌语音搜索流量的代表。

# 5.    结果

## 5.1.    与合格者一起提高质量

首先，为了提高质量，我们尝试了一致性编码器。表1给出了使用不同编码器的RNN-T模型的语音搜索测试集的比较。带有LSTM编码器的RNN-T模型作为我们的基线（B1），它被证明比传统的混合ASR系统具有更好的质量和延迟性能[1]。将[4]中的Conformer编码器限制为仅在流应用程序中使用左上下文，系统C0由于我们的多域任务中有偏差的批处理规范化统计而显示出显著的功耗降低。这可以通过随机抽样话语（即，无bucketing）来解决，而不是将长度相似的话语聚合到每个批次（表1中的C0无bucketing）中，由于每个批次的计算利用率较低，因此训练速度大大降低。通过用组标准化代替批标准化，系统C1以类似的WER增益但较少的训练速度回归解决了这个问题。进一步交换卷积模块和自我注意模块的顺序，系统C2实现了WER增益和训练加速。与B1相比，C2产生了7%的相对功耗降低和35%的加速比。注：对于B1，我们没有看到更多LSTM编码器层的质量提高。

## 5.2.    延迟改善

|  |
| --- |
| 表2：语音搜索测试集上不同RNN-T模型的质量和延迟比较。 |

在以前的工作[1,2]中，我们致力于改善RNN-T模型的端点延迟，并且已经证明E2E模型比传统的混合ASR系统具有更好的WER和端点延迟折衷。然而，即使如此，我们仍然观察到在基于E2E的流应用程序中存在较大的总延迟。这可以追溯到这样一个事实：E2E系统倾向于将部分假设延迟到端点触发的话语结束时，从而为预取留出更少的空间以减少网络延迟（表2中的B0与B1）。

### 5.2.1.    使用受限对齐减少部分延迟

在这个实验中，我们探索了用对齐信息约束RNN-T预测，如下[20]。当应用于B1时，约束对齐的使用分别实现了PR50和PR90的210ms和230ms延迟减少，质量降低的代价从6.0%降低到6.9%（B2与表2中的B1相比）。负延迟表明该模型甚至在看到演讲结束之前就能够假设整个查询。

### 5.2.2.    使用FastEmit减少部分延迟

然后我们将FastEmit应用于同一基线系统B1。从表2可以看出，B1和FastEmit（B3）给出了PR50和PR90的180ms和130ms下降，而WER回归要小得多（6.0%到6.2%）。此外，B3培训期间不需要校准。为了解决质量下降的问题，我们在表1中更好的E2E C2模型上使用了FastEmit。尽管FastEmit仍带来0.2%的绝对功率回归，但系统“C2+FastEmit”（即C3）在-110ms PR50和90ms PR90下的功率为5.8%，在质量和延迟方面都比B1更好。

### 5.2.3.    使用E2E预取降低预取延迟

接下来，我们探讨了使用E2E预取来最小化部分预取和预取之间的差距，因为预取延迟反映在总延迟中。E2E预取只能在E2E系统中使用，不适用于使用静噪预取的传统型号。从表2中可以看出，将E2E预取添加到C3（即C4）会使PF50的预取延迟降至-50ms，PF90的预取延迟降至110ms，PFR为1.86。水资源增加到6.0%，仍然与B1相当。

## 5.3.    双通道级联编码器

为了进一步解决由于第1遍中的延迟相关优化而导致的质量下降问题，首先采用了基于LAS的两遍重排序模型[1,18]，在表2中称为T1。使用可比较的端点和部分延迟度量，它将WER降低到5.3%；然而，E2E预取效率较低，因为它仅适用于RNN-tep模型。然后我们训练一个级联编码器模型（表2中的T2）。在仅因果模式下，“T2单通道偶然”的性能类似于C4；而在EOQ时，使用附加的非因果第二通道，“T2双通道”实现了与T1类似的WER，但受益于E2E预取的预取延迟要好得多。在非流应用中，我们可以使用“T2单通非因果”来获得更好的4.8%的WER。由于采用波束搜索代替重扫描，T2的计算量远高于T1。在今后的工作中，我们将进一步研究减少它的技术。与先前发布的最佳E2E系统B1相比，使用T2型双通道级联编码器，我们可以获得更好的WER和延迟（包括端点、部分和预取）。

## 5.4.    与传统模型的比较

我们还包括传统的混合ASR模型（表2中的B0）的性能，该模型由发射上下文相关音素的低帧速率声学模型[39]（0.1GB）、764个单词的发音模型（2.2GB）、第一遍5克语言模型（4.9GB）组成。与传统系统B0相比，RNN-T模型（B2除外）具有更好的质量和端点延迟，再次证实了[1]中的发现。使用FastEmit，我们可以通过非常接近的部分延迟（C3）获得更好的质量。B0的-150ms中位部分延迟（PR50）主要来自于对大量纯文本数据训练的较强LM，这种LM甚至在语音结束之前就可能假设完整查询，而E2E模型目前缺乏这种假设。然而，即使部分延迟稍差，E2E系统仍然可以在E2E预取（C4）的帮助下获得更好的预取延迟，这直接影响流asr的总延迟。通过一个额外的非偶然的第二遍（T2两遍），我们可以进一步将WER降低到5.4%，同时仍然比B0有更快的预取。

# 6.    结论

在这项工作中，我们进一步改进了流应用程序中基于RNNT的E2E模型的质量和延迟。为了提高质量，我们用最近开发的Conformer层[4]替换了编码器LSTM层，并采用了一种简单而有效的延迟正则化技术FastEmit[3]，该技术适用于任何基于传感器的模型。此外，我们采用了级联编码器[5]RNN-T模型，这两个过程共享相同的RNN-T解码器。它为我们提供了一个比以前最好的E2E系统更好更快的系统[1]，并且在质量和所有延迟指标上都超过了传统模型。

# 7.    参考文献

[1] T.N.Sainath，Y.He，B.Li，等，“一种超过服务器端传统模型质量和延迟的流式OnDevice端到端模型”，过程中。ICASSP，2020年。

[2] B.Li，S.-Y.Chang，T.N.Sainath，et al.，“实现快速准确的端到端流式ASR”，过程。ICASSP，2020年。

[3] 余建华，赵建中，李斌，等，，“快速发射：低延迟流式ASR与序列水平发射正则化”，《中国科学院学报》。ICASSP，2021年。

[4] A.Gulati，J.Qin，C.-C.Chiu，et al.，“一致性：用于语音识别的卷积增强变压器”，过程。Interspeech，2020年。

[5] A.Narayanan，T.N.Sainath，R.Pang等人，“用于统一流式和非流式ASR的级联编码器”，过程。ICASSP，2021年。

[6] 李俊杰，吴永元，高永元，等，，「大规模语音识别端对端模式之比较研究，」附件十四：2005.14327, 2020.

[7] Y.He，T.N.Sainath，R.Prabhavalkar等人，《移动设备端到端语音识别流》，过程。ICASSP，2019年。

[8] 赵国强，吴永元，等，，「序列对序列模式语音识别之研究进展」，国立台湾大学硕士论文集。ICASSP，2018年。

[9] A.Graves，“递归神经网络的序列转导”，CoRR，第abs/1211.37112012卷。

[10] W.Chan、N.Jaitly、Q.V.Le和O.Vinyals，“听、听和拼写”，更正，第abs/1508.012112015卷。

[11] S.Kim，T.Hori和S.Watanabe，“使用多任务学习的联合CTC基于注意力的端到端语音识别”，在Proc。ICASSP，2017年。

[12] N.Jaitly，D.Sussillo，Q.V.Le等，“使用部分条件作用的在线序列到序列模型”，过程。NIPS，2016年。

[13] E.Tsunoo、Y.Kashiwagi和S.Watanabe，“具有分块同步推理的流变压器ASR，”附件十四：2006.14941, 2020.

[14] C.Raffel，M.Luong，P.J.Liu等人，“通过强制单调排列实现在线和线性时间注意”，载于《美国科学院院学报》。ICML，2017年。

[15] 范国荣，周永平，陈文华，等，，“一种基于在线注意力的语音识别模型，”附件十四：1811.05247, 2018.

[16] N.Moritz、T.Hori和J.Le Roux，“引发端到端语音识别的注意”，在Proc。ICASSP，2019年。

[17] 王国荣，吴国荣，刘国生，等，，「低延迟端到端串流式语音识别与童军网路，」附件十四：2003.10369, 2020.

[18] T.N.Sainath，R.Pang，D.Rybach等人，“双通道端到端语音识别”，过程。Interspeech，2019年。

[19] D.Zhao，T.N.Sainath，D.Rybach，等，“浅层融合端到端上下文偏见”，过程。Interspeech，2019年。

[20] T.N.Sainath，R.Pang，D.Rybach等人，“端到端模型发射字计时”，过程。Interspeech，2020年。

[21]H.Inaguma，Y.Gaur，L.Lu等，“流序列到序列ASR的最小延迟训练策略”，过程。ICASSP，2020年。

[22]C.-C.Chiu，W.Han，Y.Zhang，et al.，“长格式语音识别的端到端模型比较”，Proc。ASRU，2019年。

[23]Y.Wu和K.He，“群体正常化”，在Proc。ECCV，2018年。

[24]Z.Dai，Z.Yang，Y.Yang，et al.，“Transformer XL:定长语境之外的注意语言模型”，ACL，2019。

[25]Y.Wang，A.Mohamed，D.Le，et al.，“基于变压器的混合语音识别声学建模，”附件十四：1910.09799, 2019.

[26]A.Narayanan，R.Prabhavalkar，C.-C.Chiu，et al.，“使用流端到端模型识别长格式语音”，过程。ASRU，2019年。

[27]L.A.Barroso，J.Dean和U.Holzle，¨“地球的网络搜索：谷歌集群架构”，IEEE Micro，第23卷，第22-28页，2003年。

[28]S.Jonassen、B.B.Cambazoglu和F.Silvestri，“预取查询结果及其对搜索引擎的影响”，过程。SIGIR，2012年。

[29]S.-Y.Chang，B.Li，D.Rybach等人，“使用端到端预取的低延迟语音识别”，过程。Interspeech，2020年。

[30]A.Graves、S.Fernandez、F.Gomez和J.Schmidhuber，“连接主义时间分类：用递归神经网络标记未分段的序列数据”，在Proc。ICML，2006年。

[31]M.Schuster和K.Nakajima，“日韩语音搜索”，在Proc。ICASSP，2012年。

[32]F.Biadsy、M.Ghodsi和D.Caseiro，“有效构建包含非语言信号的Tera-Scale MaxEnt语言模型”，在Proc。Interspeech，2017年。

[33]M.Abadi等人，“Tensorflow:异构分布式系统上的大规模机器学习”，在线提供：http://download.tensorflow.org/paper/whitepaper2015.pdf, 2015.

[34]J.Shen，P.Nguyen，Y.Wu，et al.，“Lingvo:一个用于序列到序列建模的模块化和可扩展框架，”附件十四：2005.08100, 2019.

[35]D.P.Kingma和J.Ba，“Adam:随机优化的方法”，在Proc。ICLR，2015年。

[36]H.Liao，E.McDermott和A.Senior，“YouTube视频转录半监督训练数据的大规模深度神经网络声学建模”，过程。ASRU，2013年。

[37]C.Kim，A.Misra，K.Chin，et al.，“在虚拟房间中生成大规模模拟话语，以便在Google Home中训练用于远场语音识别的深层神经网络”，Proc。Interspeech，2017年。

[38]J.Li，D.Yu，J.Huang，和Y.Gong，“利用CD-DNN-HMM中的混合带宽训练数据改进宽带语音认知”，过程。SLT，2012年。

[39]G.Pundak和T.N.Sainath，“低帧速率神经网络声学模型”，过程。Interspeech，2016年。

[[1]](" \l "_ftnref1" \o ")同等贡献