用于语音转文本任务的混合式转导器与注意力编码器-解码器建模

Yun Tang[⋆], Anna Y. Sun[⋆], Hirofumi Inaguma[⋆], Xinyue Chen[♢,*], Ning Dong[⋆], Xutai Ma[⋆], Paden D. Tomasello[⋆], Juan Pino[⋆]

Meta AI[⋆]、卡内基梅隆大学[♢]
{yuntang,hirofumii}@meta.com

arXiv:2305.03101v1 [cs.CL] 4 May 2023

摘要

转导器和基于注意力的编码器-解码器（AED）是语音转文本任务中两种广泛使用的框架。它们针对不同目的而设计，在语音转文本任务中各有优缺点。为充分发挥这两种建模方法的优势，我们提出一种将转导器与基于注意力的编码器-解码器（TAED）相结合的语音转文本任务解决方案。该新方法利用 AED 在非单调序列到序列学习方面的优势，同时保留转导器的流式处理特性。在所提出的框架中，转导器和 AED 共用同一个语音编码器。转导器中的预测器由 AED 模型中的解码器取代，解码器的输出以语音输入为条件，而非以无条件语言模型的输出为条件。所提出的方案确保模型通过涵盖所有可能的读写场景进行优化，并为流式应用创建匹配的环境。我们在 MUST-C 数据集上评估了所提出的方法，结果表明，在离线自动语音识别（ASR）和语音翻译（ST）任务中，TAED 的表现显著优于转导器。在流式场景下，TAED 在 ASR 任务和一个语音翻译方向上优于转导器，而在另一个翻译方向上则取得了相当的结果。

1 引言

基于神经网络的端到端框架在语音转文本任务中取得了显著成功，例如自动语音识别（ASR）和语音到文本翻译（语音翻译）（Li，2021年）。这些框架包括基于注意力的编码器-解码器建模（注意力编码器-解码器）（Bahdanau 等人，2014）、连接时序分类（CTC）（Graves 等人，2006年）和转导器（Graves，2012）等。它们针对不同的目的进行设计，具有截然不同的行为，尽管它们都可用于解决从语音输入序列到文本输出序列的映射问题。

注意力编码器-解码器通过让解码器关注源序列的部分内容来实现序列到序列学习。它提供了一种强大且通用的解决方案，不受输入/输出模态、长度或序列顺序的限制。因此，它广泛应用于自动语音识别（Chorowski 等人，2015年；Chan 等人，2015年；Zhang 等人，2020年；Gulati 等人，2020年；Tang 等人，2021年）和语音翻译（Berard 等人，2016；Weiss 等人，2017年；Li 等人，2021年；Tang 等人，2022）。

连接时序分类及其变体转导器旨在处理语音输入序列与文本输出序列之间的单调对齐。解码过程中会在语音特征与目标文本词元之间生成硬对齐，其中每个输出词元都与一个输入语音特征相关联或同步。连接时序分类和转导器具有许多适用于自动语音识别的理想特性。例如，它们天然适用于流式应用，而输入同步解码有助于缓解注意力编码器-解码器中的过度生成或生成不足问题。Sainath 等人（2019年）；Chiu 等人（2019年）表明，在长话语识别中，转导器的词错误率优于注意力编码器-解码器，而在短话语场景中，注意力编码器-解码器的表现则优于转导器。另一方面，研究表明，连接时序分类和转导器在处理非单调序列映射时并非最优（Chuang 等人，2021年），尽管一些初步尝试已取得令人鼓舞的进展（Xue 等人，2022；Wang 等人，2022）。

在这项工作中，我们提出了一种混合转导器与AED模型（TAED），将AED模型和转导器模型整合到同一个框架中，以充分发挥两种建模方法的优势。在TAED中，AED与转导器共享语音编码器。转导器中的预测器

*Xinyue Chen 在 Meta 实习期间参与了这项工作。

由AED中的解码器取代。AED解码器的输出辅助转导器的连接器预测输出词元。训练期间，转导器模型与AED模型受到同等对待并进行联合优化，而推理时仅使用转导器连接器的输出。我们将TAED模型扩展到基于块的同步方案下的流式应用，该方案确保训练集涵盖所有读写选择，并消除训练与推理之间的不一致。我们研究了流式延迟与AED对齐之间的关系，并提出一种简单、快速的AED对齐方法，以较小的质量损失实现低延迟。随后，我们在离线和流式设置下对这一新方法进行了自动语音识别和语音翻译任务评估。结果表明，该方法有助于在离线评估中取得新的最先进结果。相应的流式扩展也能在相近的延迟预算下显著提升质量。总而言之，我们的贡献如下：

1. TAED，即转导器和注意力编码器-解码器建模的混合方法，被提出用于语音转文本任务

2. 采用基于块的流式处理同步方案，以消除流式处理应用中的训练与推理差异。

3. 采用简单、快速的 AED 对齐，以平衡 TAED 的延迟和质量。

4. 所提出的方法在自动语音识别和语音翻译任务的离线和流式处理设置下均取得了最先进水平的结果。

2 预备知识

形式上，我们将一个语音转文本任务的训练样本表示为一对（$\mathbf{x}$,$\mathbf{y}$）。$\mathbf{x} = x_{1:T}$和$\mathbf{y} = y_{1:U}$分别是语音输入特征和目标文本词元。T和U分别是对应的序列长度。$y_u \in \mathcal{V}$和$\mathcal{V}$是目标词汇表。目标函数是在训练集上最小化负对数似然 log p($\mathbf{y}$|$\mathbf{x}$ θ)

$$\mathcal{L}_{\text{aed}} = -\sum_{(\mathbf{x},\mathbf{y})} \log p(y_u|y_{1:u-1}, x_{1:T}). \quad (1)$$

在流式处理设置中，模型会在时间戳 a =(t_1、$\cdots$、t_u、$\cdots$、t_U) 处生成预测，而不是等到话语结束时，其中 $t_u \le t_{u+1}$ 和 $0 < t_u \le T$。我们将预测时间戳序列称为语音 $x_{1:T}$ 与词元标签$y_{1:U}$之间的对齐 a。$\mathcal{A}_T^U = \{a\}$ 表示$x_{1:T}$与 $y_{1:U}$之间的所有对齐。流式模型参数 θ_{s}通过

$$\min_{\theta_{\text{s}}} \sum_{(\mathbf{x},\mathbf{y})} \sum_{a \in \mathcal{A}_T^U} \sum_{u=1}^{U} -\log p(y_u|y_{1:u-1}, x_{1:t_u}). \quad (2)$$

离线建模可以被视为流式建模的一种特殊情况，即对所有 $t_u = T$ 而言，对齐是唯一的。以下两个小节将简要介绍我们混合方法中使用的两种建模方法。

2.1 基于注意力的编码器-解码器

注意力编码器-解码器（AED）由一个编码器、一个解码器和注意力模块组成，这些注意力模块将编码器和解码器中的对应层连接起来，如图 1(a) 所示。编码器根据输入 $h_{1:T}$ 生成上下文表示 $x_{1:T}$ [1]。解码器状态 s_u^l 根据先前的状态和编码器输出进行估计

$$s_u^l = f_{\theta_{\text{dec}}}(h_{1:T}, s_{1:u-1}^l, y_{u-1}), \quad (3)$$

其中，$f_{\theta_{\text{dec}}}$ 是由 θ_{dec} 参数化的神经网络，$l \in [1, L]$ 是层索引。

当注意力编码器-解码器扩展到流式处理应用时（Raffel 等人，2017年；Arivazhagan 等人，2019年），一个关键问题随之产生：

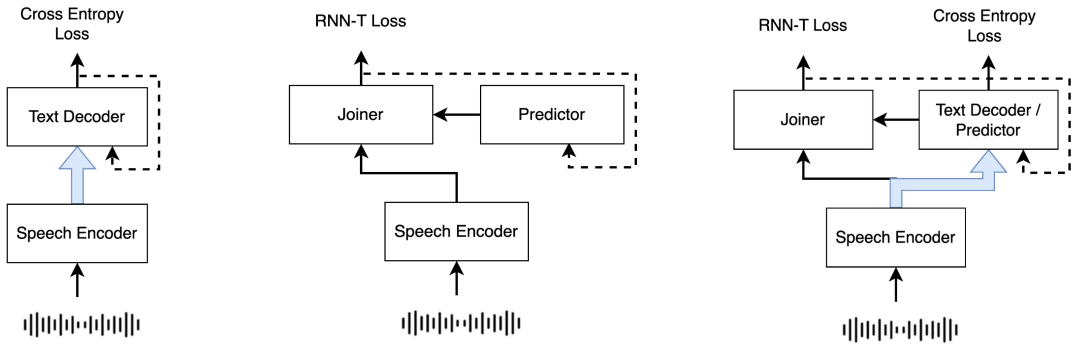
我们如何确定解码器的写入/读取策略？

假设 AED 模型基于 Transformer，且在推理过程中，令牌 $y_{1:u-1}$ 已在时间步 t 之前解码。下一个 AED 解码器状态 $s_u^l(t)$ 与部分语音编码器输出 $h_{1:t}$ 以及部分对齐 $a' \in \mathcal{A}_t^{u-1}$相关，后者位于 $h_{1:t}$ 和 $y_{1:u-1}$之间。Transformer 解码器层的计算（Vaswani 等人，2017年）包括一个自注意力模块和一个交叉注意力模块。自注意力模块对先前解码器状态中的相关信息进行建模

$$\hat{s}_{a'}^l = [s_1^l(t_1), \cdots, s_{u-1}^l(t_{u-1})], \quad (4)$$

其中，t_{u-1} 是对齐 a' 中 token$u-1$ 的预测时间戳。交叉注意力模块

[1]语音编码器中可能会应用下采样模块。为简单起见，我们仍使用 T作为编码器输出序列长度。



（a）基于注意力的编码器-解码器。

（b）转导器

（c）混合转导器与基于注意力的编码器-解码器。

图 1：不同框架的比较：注意力编码器-解码器、转导器和 TAED。

从编码器输出 $h_{1:t}$ 中提取信息。解码器状态的计算方式修改为

$$s_u^l(t) = f_{\theta_{\text{dec}}}(h_{1:t}, \hat{s}_{a'}^l, y_{u-1}). \tag{5}$$

为覆盖训练期间所有的读写路径，我们需要在给定输出词元序列 $y_{1:U}$ 的每个时间步枚举所有可能的对齐方式 O（$T'!\ \ (T'-U)\ !\ U!$），其计算成本高得难以承受。在基于注意力编码器-解码器的方法中，例如单调无限回看注意力（MILk）（Arivazhagan 等人，2019年）和单调多头注意力（MMA）（Ma 等人，2020b），使用上下文向量的估计值来避免枚举对齐方式。在交叉注意力增强转导器（CAAT）（Liu 等人，2021年）中，连接器中的自注意力模块被移除，以实现解耦 $y_{1:u-1}$ 和 $h_{1:t}$。

2.2 转导器

转导器有三个主要组成部分。语音编码器 θ_{enc} 构建上下文语音表示 $h_{1:T}$，该表示来自语音输入 $x_{1:T}$；预测器 θ_{pred} 根据先前的目标词元对语言信息进行建模，而连接器θ_{joiner} 则融合声学表示和语言表示，为每个语音输入特征预测输出，如1(b)所示。编码器和预测器通常采用循环神经网络（RNN）（Graves，2012）或 Transformer（Zhang 等人，2020年）架构进行建模。连接器模块是一个前馈网络，它将语音编码器 h_t 和预测器的输出 s_u^L 扩展为一个维度为$T \times U$的矩阵（$z(t,\ u)$:

$$z(t,u) = f_{\theta_{\text{joiner}}}(h_t, s_u^L). \tag{6}$$

对 $z(t,\ u)$ 应用线性投影 $W^{\text{out}} \in \mathcal{R}^{d \times |\mathcal{V} \cup \varnothing|}$，以获得每个输出词元的 logits$k \in \mathcal{V} \cup \varnothing$。如果非空白词元与当前 h_t不匹配，则生成空白标记 $\varnothing$。RNN-T 损失使用前向-后向算法进行优化：

$$\begin{aligned}\alpha_{t,u} = \text{LA}\Big(&\alpha_{t,u-1} + \log p\big(y_u|z(t,u-1)\big),\\ &\alpha_{t-1,u} + \log p\big(\varnothing|z(t,u)\big)\Big),\end{aligned} \tag{7}$$

$$\mathcal{L}_{\text{rnn-t}} = -\alpha_{T,U} - \log p(\varnothing|T,U), \tag{8}$$

其中 LA($x,\ y$) $=\log(\exp^x + \exp^y)$，且 $\alpha_{0,\phi}$ 初始化为 0。转导器非常适合流式任务，因为它可以从数据中隐式学习读写策略，即空白标记表示读取操作，非空白标记表示写入操作。

3 方法

在本研究中，我们选择 Transformer-转导器（T-T）（Zhang 等人，2020年）作为所提出 TAED 系统的骨干网络。对于流式处理设置，语音编码器采用分块实现（Chiu 和 Raffel，2017年；Chen 等人，2020年），按块大小 N 接收并计算新的语音输入数据，而不是每次处理一帧。

3.1 TAED

TAED将转导器和注意力编码器-解码器整合到一个模型中，如图1(c)所示。转导器和注意力编码器-解码器模型共享语音Transformer编码器。转导器中的预测器由AED解码器取代。其输出为

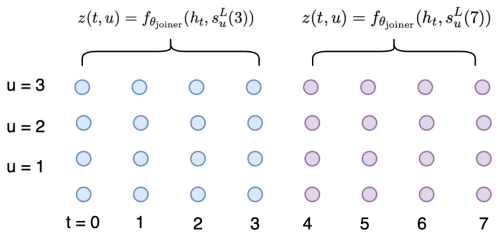


图2：基于分块的 RNN-T 同步。来自 $t \in [0,3]$ 的帧属于第一个块（青色），而 $t \in [4,7]$ 属于第二个块（紫色）。

新预测器的输出同时包含语音编码器输出和预测词元的结果，因此能为连接器提供更多信息。

转导器和AED模型通过两个准则联合优化：针对转导器连接器输出的RNN-T 损失，以及针对AED解码器输出的交叉熵损失。总损失 $\mathcal{L}_{taed}$是这两个损失的总和。

$$\mathcal{L}_{\text{tead}} = \mathcal{L}_{\text{rnn-t}} + \mathcal{L}_{\text{aed}}. \tag{9}$$

该模型基于转导器连接器的输出进行评估。

3.2 基于块的 RNN-T 同步方案中的转导器优化

当我们尝试将 TAED 扩展到流式处理场景时，会遇到 §2.1 中讨论过的相同流式读写问题。为了避免枚举数量呈指数级增长的对齐方式，我们采用了一种不同的方法，并修改推理逻辑，使其与训练和推理条件相匹配。

在传统的流式解码器推理中，当新的语音编码器输出 h_t 可用时，新的解码器状态 $s_u^l(t)$ 通过 $h_{1:t}$ 以及之前计算得到的解码器状态 $\hat{s}_{a'}^l$ 进行估计，这些状态基于 $h_{1:t'}$ 和 $t' \leq t$，如公式（5）所示。在所提出的解决方案中，我们根据语音编码器输出 $h_{1:t}$更新所有之前的解码器状态，并将 $\hat{s}_{a'}^l$ 替换为 $\hat{s}_{a(t)}^l$。

$$\hat{s}_{a(t)}^l = [s_1^l(t), \cdots, s_{u-1}^l(t)], \tag{10}$$

其中，$a(t)$ 表示一种特殊对齐，所有词元都对齐到时间戳 $t_u = t$。

这一修改背后有两个原因。首先，我们预计，当有更多语音数据可用时，更新所有解码器状态会使状态表示更加准确。其次，这一修改有助于减少训练期间 $y_{1:U}$ 与 $h_{1:t}$ 之间的大量对齐，降至一个，即 $a(t)$。与传统的注意力编码器-解码器训练相比，这只会使解码器前向计算增加 T 倍。

使用基于分块的编码器时，计算成本会进一步降低。给定两个解码器状态 $s_u^l(t)$和 $s_u^l(\delta(t))$，其中 $\delta(t)$是 t 所属块的最后一帧索引，并且 $t \leq \delta(t)$，$s_u^l(\delta(t))$比$s_u^l(t)$包含更多信息，因为前者接收了更多的语音输入。在推理过程中，当语音块数据可用时，$s_u^l(t)$和 $s_u^l(\delta(t))$会同时可用。因此，在推理和训练中，我们都将所有 $s_u^l(t)$替换为相应的 $s_u^l(\delta(t))$。如果 N 是一个语音输入块中的语音编码器输出帧数，那么基于分块的计算有助于在训练期间将解码器计算成本降低 N 倍，因为我们只需每 N 帧更新一次解码器状态，而不是每帧更新一次。总之，$s_u^l(t)$的计算由公式（5）修改如下

$$s_u^l(t) = f_{\theta_{\text{dec}}}(h_{1:\delta(t)}, \hat{s}_{a(\delta(t))}^l, y_{u-1}), \tag{11}$$

以及连接器输出 $z(t,u)$ 在公式（6）中更新为

$$z(t,u) = f_{\theta_{\text{joiner}}}(h_t, s_u^L(t)). \tag{12}$$

基于分块的 RNN-T 同步如图2所示。一个块中的语音编码器输出帧数为 4。$z(t,u)$s 在第一个块（青色）中通过 s_u^L（3）计算，第二个块（紫色）则基于 s_u^L（7）。

当块大小 N 长于语句长度 T时，基于分块的 TAED 流式训练与具有相似计算成本的转导器离线训练相同。

3.3 AED优化与快速对齐

除了使用上述 RNN-T 损失优化模型外，我们还为 AED 模型引入了另一种辅助损失 $\mathcal{L}_{\text{aed}}$，如公式（9）所示。一种直接的方法是按照公式（1）将 AED 模块作为离线 AED 模型进行优化。然而，如果将离线 AED 模型用于流式应用，可能会导致较高的延迟。因此，我们还通过创建一种针对语音编码器输出的目标词元均匀对齐方式 a^e，引入了一种简单的“流式”模式 AED 训练，即 $t_u^e = \lfloor u * T'/U \rfloor$，其中 $\lfloor x \rfloor$ 是对 x执行向下取整运算。此外，我们还可以通过对齐加速因子来调整对齐速度

$\lambda > 0$，而新的对齐 a^{λ} 是使用时间步 $t_u^{\lambda} = \max(T, \lfloor \frac{u*T'}{U*\lambda} \rfloor)$。当 $\lambda > 1.0$时，流式 AED模型使用快速对齐进行训练，并鼓励利用更少的语音数据预测新词元。另一方面，如果 $\lambda < \frac{1}{U}$，那么 $t_u^{\lambda} = T$，这等同于离线训练。辅助 AED 任务通过

$$\mathcal{L}_{\text{aed}} = -\sum_{(\mathbf{x},\mathbf{y})} \sum_{u} \log p(y_u | y_{1:u-1}, h_{1:t_u^{\lambda}}). \quad (13)$$

请注意，这种方法不需要精确对齐，而在与翻译相关的应用中，精确对齐可能难以获得。

3.4 推理期间的空白惩罚

输入语音帧数与目标词元数之间的比例可能因诸多因素而异，例如语速不同或目标语言单位不同。同时，预测空白标记 $\varnothing$ 表示读取操作，而非空白词元表示写入操作，如第 2.2 节所述。在推理期间，引入空白惩罚 τ，通过惩罚空白标记$\varnothing$的发射概率来调整目标词元的产出率。它的作用类似于自动语音识别（Takeda 等人，1998年）中使用的词插入惩罚：

$$\hat{e}(t,u)[i_{\varnothing}] = e(t,u)[i_{\varnothing}] - \tau, \quad (14)$$

其中 $e(t, u) = \text{LOGSOFTMAX}_u(W^{\text{out}} z(t, u))$，且 $i_{\varnothing}$ 是空白标记的索引 $\varnothing$。

4 流式处理算法比较

何时读取新输入并写入新输出，是流式处理算法中的一个基本问题。根据流式处理读写策略的选择，大致可分为两类：预先固定策略和自适应策略。预先固定策略（如 Wait-k (Ma 等人，2019年）采用固定方案读取新输入并写入新输出。另一方面，自适应算法会根据当前输入的语音数据动态选择读写策略。根据输入与输出的同步方式，自适应算法还可进一步分为两类。

第一类自适应流式处理算法基于 AED 框架，包括硬单调注意力（HMA）（Raffel 等人，2017年）、MILk（Arivazhagan 等人，2019年）、MoChA（Chiu 和 Raffel，2017年）、MMA（Ma 等人，2020b）以及连续积分发放（CIF）（Dong and Xu，2020年；Changandyi Lee，2022）。这些方法通过编码器与解码器之间的注意力机制，从编码器输出中提取声学信息。解码器将声学信息与语言信息融合；语言信息则根据已解码的词元历史估计得出。输入序列与输出序列之间没有显式对齐；换言之，输出与输入**不同步**。如 §2.1所述，AED 模型难以直接用于流式处理，因此训练期间采用了近似方法。例如，通过编码器与解码器之间的注意力提取的、依赖对齐的上下文向量，通常会被根据对齐计算出的上下文向量期望所取代。这与推理过程不同，推理基于解码期间采样得到的特定对齐路径。因此，训练与推理之间不可避免地存在差异，可能会损害流式处理性能。

第二类自适应流式处理方法具有**同步的**输入和输出，其中每个输出标记都与一个语音输入帧相关联。这包括连接时序分类、Transducer、CAAT 以及所提出的 TAED。如果采用语言建模，它们会在连接器中结合声学和语言信息。训练期间不需要进行特定的读写决策。该方法考虑所有对齐，并通过连接时序分类损失或 RNN-T 损失进行优化。因此，不存在训练和推理之间的不一致。不同方法的详细比较列于表1中。

5 实验

5.1 实验设置

数据 实验是在两个 MuST-C（Gangi 等人，2019年）语言对上进行的：英语到德语（EN→DE）和英语到西班牙语（EN→ES）。采用序列级知识蒸馏（Kim 和 Rush，2016）来提升语音翻译（ST）的质量（Liu 等人，2021年。EN→ES方向的数据中英语部分用于英语自动语音识别（ASR）的开发和评估。模型在开发集 上进行开发，最终结果则在`tst-COMMON`集上报告。此外，为便于与其他ASR系统进行比较，我们还在附录 C 中报告了 LIBRISPEECH（Panayotov 等人，2015）的ASR结果。

评估 自动语音识别质量以词错误率（WER）衡量，语音翻译质量则以区分大小写、去分词后的 BLEU 报告，后者】【。

方法	同步	合并模块	读写决策	训练/推理
CIF（Dong andXu，2020年）	异步	解码器	$h_{\leq j}$	采样+缩放/采样
HMA（Raffel 等人，2017年）	异步	解码器	$h_j, s_{\leq i}$	期望/采样
MILk (Arivazhagan 等人, 2019年)	异步	解码器	$h_{\leq j}, s_{<i}$	期望/采样
MMA（Ma 等人，2020b）	异步	解码器	$h_{\leq j}, s_{<i}$	期望/采样
连接时序分类（Graves 等人，2006年）	Sync	None	h_j	所有路径/采样
转导器（Graves，2012）	Sync	连接器	$h_j, s_{k<i}$	所有路径/采样
CAAT(Liu 等人，2021年)	Sync	连接器	$h_{\leq j}, s_{<i}$	所有路径/采样
TAED（本文工作）	Sync	预测器，连接器	$h_{\leq j}, s_{<i}$	所有路径/采样

表 1：不同流式处理方法的比较。“读写决策”列列出了进行读写决策所需的信息。

基于默认的 SACREBLEU 选项（Post，2018年）[2]。延迟通过平均滞后（AL）（Ma 等人，2019年）使用 SimualEval（Ma 等人，2020a）进行测量。

模型配置 输入语音表示为 80 维对数梅尔滤波器组系数，每 10 毫秒计算一次，使用 25 毫秒窗口。采用全局通道均值和方差归一化。在所有实验中，均采用 LB 策略的 SpecAugment（Park 等人，2019年）数据增强。目标词汇表由 1000 个“单字”子词单元组成，这些单元通过 SentencePiece（Kudo 和 Richardson，2018年）学习得到，覆盖所有训练文本数据中的全部字符。

我们选择 Transformer 转导器（T-T）（Zhang 等人，2020年）作为转导器基线模型。语音编码器由两个因果卷积层构成，卷积核大小为 3，步幅大小为 2。输入语音特征先进行 4 倍下采样，再经过 16 个采用相对位置嵌入的分块式 Transformer 层（Shaw 等人，2018年）。在流式处理场景下，语音编码器可以访问当前时间步之前所有块中的语音数据，以及当前时间步之后的一个块（Wu 等人，2020年；Shi 等人，2020年；Liu 等人，2021年）。我们将块大小设为 160 毫秒至 640 毫秒进行遍历。对于离线模型，按照 §3.2所述，我们只需将块大小设为大于待处理的任何话语。预测器模块包含两个 Transformer 层。语音编码器和预测器中的 Transformer 层，其输入嵌入维度均为 512，注意力头数为 8，中间层维度为 2048。连接器模块与 T-T（Zhang 等人，2020年）一样，是一个前馈神经网络。TAED 与 T-T 基线采用相同的配置，不同之处在于，预测器模块被 AED解码器取代，并增加了用于连接输出与语音编码器输出的注意力模块

[2]大小写混合+数字引用.?+平滑指数+无分词器+版本.1.5.1

模型	BLEU（↑）	
	英语→德语	英语→西班牙语
注意力编码器-解码器（Wang 等人，2020年）	22.7	27.2
注意力编码器-解码器（Inaguma 等人，2020年）	22.9	28.0
CAAT（Liu 等人，2021年）	23.1	27.6
转导器	24.9	28.0
TAED	**25.7**	**29.6**

表 2：MUST-C tst-COMMON 集上的离线语音翻译对比。

。对于转导器和 TAED 配置，参数总数均约为 5900 万。

超参数设置 模型使用 T-T 架构，通过自动语音识别任务进行预训练。训练好的语音编码器用于初始化 TAED 模型和基于 T-T 的语音翻译模型。模型使用 16 个 A100 GPU 进行微调，最多更新 300k 次。每个 GPU 的批量大小为 16k 个语音帧。离线模型约需一天训练完成，流式模型则需三天，这是因为前瞻块和基于块的同步方案带来了额外开销。如果连续 20 个周期训练没有进展，则采用提前停止策略。所有实验均采用学习率为 3e-4 的 RAdam 优化器（Liu 等人，2020 年）。标签平滑率和 dropout 率均设为 0.1。我们通过网格搜索选择空白惩罚项 τ，搜索范围为 [0, 4。0]，步长为=0.5，在 开发集上进行。模型使用 FAIRSEQ (Wang 等人，2020年) 训练。推理时对效果最佳的十个检查点取平均，并采用贪心搜索（束宽=1）。

5.2 离线结果

离线模型的结果列于表2 和表3中。在表2中，我们将模型与仅使用 MUST-C 数据报告的系统进行了比较。前两行基于 AED 框架，第三行则是基于 CAAT 的结果，而 CAAT 是 IWSLT2021（

模型	词错误率 (↓)	
	dev	tst-COMMON
转导器	14.3	12.7
TAED	**11.9**	**10.9**

表 3：MUST-C 开发集和 tst-COMMON 测试集上的离线自动语音识别对比。

Anastasopoulos 等人，2021年）和 IWSLT2022（Anastasopoulos 等人，2022）流式获胜系统的核心。我们的转导器基线取得了具有竞争力的结果，与上述三个系统相当。切换到所提出的 TAED 框架后，BLEU 提高了 0.8 至 1.6。表 3 展示了相应的 ASR 质量；在 tst-COMMON集上，与转导器基线相比，TAED 实现了 14% 的相对 WER 降低。

结果表明，在语音翻译任务中，转导器能够取得与基于注意力编码器-解码器（AED）的模型具有竞争力的结果。以语音编码器输出为条件的预测器能够为连接器提供更准确的表示。TAED 能够兼顾转导器和 AED 的优势，并取得更好的结果。

在下一个实验中，我们比较 AED 任务权重对离线模型的影响。在公式（9）中，RNN-T 损失和 AED 交叉熵损失相加，构成训练过程中的总损失。在表 4中，我们将训练期间的 AED 任务权重从 0.0 调整到 2.0。第 2、3、4 列分别对应 AED 任务权重、ASR 词错误率，以及“EN→ES”方向上的 ST BLEU。AED 权重为 0.0 表示仅使用 RNN-T 损失，而 AED 权重 = 1.0 则等同于公式（9）中提出的方法。即使没有 AED 任务提供额外指导（AED 权重=0.0），模型在 ASR 和 ST 任务上的表现仍优于 Transducer 模型，不过提升幅度减半。在训练期间引入 AED 任务，即 AED 权重大于 0 时，采用 0.5、1.0 和 2.0 这三种 AED 权重都取得了相近的结果。这表明 AED 指导至关重要，而且任务权重对最终结果并不十分敏感。在接下来的流式实验中，我们将遵循公式（9），不改变 AED 任务权重。

模型	注意力编码器-解码器权重	词错误率（↓）	BLEU（↑）
转导器	–	12.7	28.0
TAED	0.0	11.9	28.9
	0.5	10.9	30.1
	1.0	10.9	29.6
	2.0	**10.8**	**30.2**

表 4：在 MUST-C tst-COMMON 集上使用不同 AED 权重训练的 TAED 模型比较。“BLEU”表示“EN→ES”方向上的 ST 结果，“WER”列包含相应的 ASR 结果。

5.3 流式结果

我们首先研究 §3.3 中所述 AED 对齐加速因子的影响，结果见表 5 和表 6。这些实验中，块大小设为 320 毫秒。自动语音识别结果见表 5。第一行表示对齐加速因子 λ。“Full”表示将 AED模型作为离线语音翻译模型进行训练。“1.0”表示沿时间轴均匀分布标记所创建的对齐。使用离线 AED模型（“Full”）训练的流式处理 TAED模型，其词错误率为 12.7，但延迟较高。我们检查了解码结果，发现模型倾向于在输入话语接近结束时才生成第一个非空白标记。连接器学会了等待到话语末尾再生成可靠输出，并倾向于忽略直接语音编码器输出。采用快速AED对齐后，即 $\lambda \geq 1.0$，延迟从近 6 秒显著降低至不到 1 秒。越大，平均滞后越小。一个令人意外的发现是，随着 λ 增大，词错误率和延迟都会降低。随着 λ 从 1.0 增至 1.2，词错误率从 14.7 降至 12.5，略优于使用离线 AED模块训练的 TAED。我们推测，如果连接器能同时从语音编码器输出和 AED 解码器输出中获得同步信号，就可能取得更好的结果。当 λ 较小时（即 1.0），AED解码器输出在用于预测下一个标记时可能会滞后于语音编码器输出。

表6 所示，语音翻译中也有类似发现：快速的 AED 对齐有助于降低 TAED 的延迟，不过使用离线 AED 模块时 BLEU 得分最高。与使用离线 AED 模块训练的 TAED 模型相比，两个翻译方向的延迟均从4+秒降至不到 1.2 秒，但 BLEU 得分有所下降。

λ	Full	1.0	1.2	1.4
词错误率（↓）	12.7	14.7	**12.5**	12.7
平均滞后（↓）	5894	1654	**849**	907

表 5：AED 对齐加速因子对 MUST-C EN tst-COMMON 测试集上流式自动语音识别性能的影响比较。

λ	EN→ES		EN→DE	
	BLEU（↑）	平均滞后（↓）	BLEU（↑）	平均滞后（↓）
Full	**28.3**	4328	**24.1**	4475
1.0	27.1	1715	22.8	1611
1.2	27.6	1228	22.6	1354
1.4	27.6	**1120**	22.3	**1208**

表 6：AED 对齐速度提升系数对 MUST-C tst-COMMON 数据集上流式语音翻译性能的影响比较。对于 EN→ES 和 EN→DE，我们均将块大小设置为 320 毫秒。

从 0.9（EN→ES）降至 1.8（EN→DE）。

在接下来的实验中，我们比较 TAED 和转导器的质量与延迟。通过将块大小从 160、320 和 480 毫秒改为 640 毫秒，我们构建了具有不同延迟的模型。我们在图3中给出了词错误率与平均滞后的关系曲线。虚线表示离线模型的词错误率，实线表示流式模型的词错误率。图中显示，与相应的转导器模型相比，所提出的 TAED 模型取得了更低的词错误率，绝对降幅为 1.2 至 2.1，且延迟相近。

ST 的 BLEU 与 AL 曲线如图4 和图5 所示，分别对应 EN→ES 和 EN→DE 方向。除了 Transducer 和 TAED 的结果外，我们还纳入了 Liu 等人(2021年) 的 CAAT 结果，以便于比较。首先，在 EN→ES 方向的不同操作点上，TAED 始终优于 Transducer；在 EN→DE 方向上，两者表现相当。我们预计，当给予更多延迟预算时，TAED 模型在 EN→DE 方向上会优于 Transducer 模型，因为离线 TAED 模型明显优于相应的离线 Transducer 模型。其次，CAAT 在极低延迟区域（∼ 1 平均滞后秒数）表现更好；当 EN→ES 方向的平均滞后超过 1.1 秒、EN→DE 方向超过 1.3 秒时，TAED 开始优于 CAAT。当两个方向的延迟都超过 1.4 秒时，TAED 的 BLEU 分数高于离线 CAAT 模型。详细结果见附录 B。

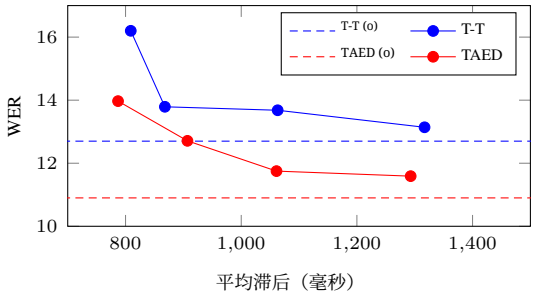


图 3：MUST-C 英语 tst-COMMON 数据集上词错误率（WER，↓）与平均滞后（↓）的关系（$\lambda = 1.4$）。“o”表示离线模型。

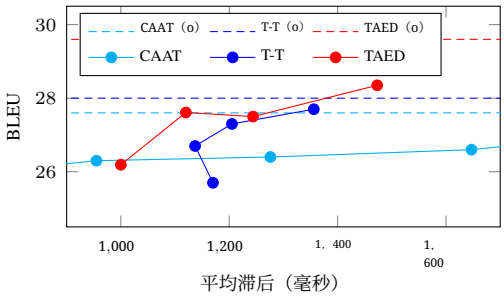


图 4：MUST-C EN→ES tst-COMMON 数据集上的 BLEU (↑) 与平均滞后 (↓) ($\lambda = 1.4$)

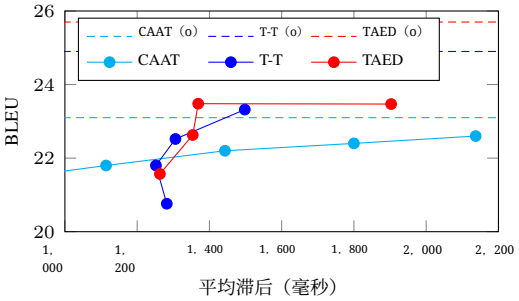


图 5：MUST-C EN→DE tst-COMMON 数据集（$\lambda = 1.2$）上的 BLEU（↑）与平均滞后（↓）

6 相关工作

鉴于注意力编码器-解码器和连接时序分类/转导器各自的优势与不足，许多研究致力于将这些方法结合起来。

带注意力的转导器（Prabhavalkar 等人，2017年）也是一种转导器变体，它同样将编码器输出馈送给预测器。我们的方法在两个方面有所不同。首先，我们将 TAED 视为两个不同模型的组合：转导器和注意力编码器-解码器。训练期间，两者以相等的权重进行优化，而带注意力的转导器仅使用 RNN-T 损失进行优化。如表 4 所示，这对于取得具有竞争力的结果至关重要。其次，我们的方法还包含一种流式处理方案，而带注意力的转导器只能应用于离线建模。

另一种解决方案是通过两遍方法结合这两种方法（Watanabe 等人，2017年；Sainath 等人，2019年；Moriya 等人，2021年）。第一遍通过束搜索获得一组完整假设。第二遍模型通过结合两个模型的似然分数对这些假设重新评分，并返回得分最高的结果。这一研究方向上的改进是用单遍解码取代两遍解码，在束搜索过程中将连接时序分类/转导器的分数与注意力编码器-解码器结合起来（Watanabe 等人，2017年；Yan 等人，2022）。然而，由于两种方法之间存在同步差异，因此需要复杂的解码算法。这也会导致较高的计算成本和延迟（Yan 等人，2022）。此外，两遍方法并不天然适用于流式应用。研究中采用了触发式解码等启发式方法（Moritz 等人，2019年；Moriya 等人，2021年）。在我们提出的解决方案中，两个模型紧密集成，并原生支持流式处理；TAED 的预测是两个模型协同作用的结果。

7 结论

在本研究中，我们提出了一种新框架，将转导器和注意力编码器-解码器模型整合到一个模型中。这种新方法确保优化覆盖所有读写路径，并消除流式应用中训练与评估之间的差异。TAED在离线自动语音识别和语音翻译任务中取得了优于主流注意力编码器-解码器和转导器建模方法的结果。在流式场景下，TAED模型在英语自动语音识别任务和EN→ES语音翻译任务中均持续优于转导器基线，同时在EN→DE方向上取得了相当的结果。此外，在中等和较高延迟范围内，其性能也优于最先进的流式语音翻译系统（CAAT）。

8 局限性

由于需要通过注意力模块连接语音编码器和 AED 解码器，TAED模型的参数略多于对应的转导器模型。离线模型的训练时间相近。然而，由于采用了 §2.1 中所述的基于分块的 RNN-T 同步方案，流式模型的优化需要更多 GPU 显存和计算时间。在我们的实验中，流式 TAED 模型的训练耗时约为离线模型的三倍，使用 16 张 A100 GPU 卡，每张 GPU 显存为 40GB。

在本研究中，我们在两个翻译方向上评估了流式处理语音翻译算法：EN→ES 和 EN→DE。英语和西班牙语的词序为主语-动词-宾语（SVO），而德语的词序为主语-宾语-动词（SOV）。实验验证了这些流式处理算法在词序不同和词序相似的语言对上的效果。未来，我们将把研究扩展到英语以外的其他源语言以及更多语言方向。

参考文献

Antonios Anastasopoulos、Loïc Barrault、Luisa Ben-tivogli、Marcely Zanon Boito、Ondřej Bojar、Roldano Cattoni、Anna Currey、Georgiana Dinu、Kevin Duh、Maha Elbayad 等。2022。IWSLT 2022 评估活动的研究成果。发表于国际口语语言翻译研讨会。

Antonios Anastasopoulos、Ondrej Bojar、Jacob Bremerman、Roldano Cattoni、Maha Elbayad、Marcello Federico、Xutai Ma、Satoshi Nakamura、Matteo Negri、Jan Niehues、Juan Miguel Pino、Elizabeth Salesky、Sebastian Stüker、Katsuhito Sudoh、Marco Turchi、Alexander H. Waibel、Changhan Wang 和 Matthew Wiesner。2021年。IWSLT 2021年评估活动的发现。发表于国际口语翻译研讨会。

N. Arivazhagan、Colin Cherry、Wolfgang Macherey、Chung-Cheng Chiu、Semih Yavuz、Ruoming Pang、Wei Li 和 ColinRaffel。2019年。用于同步机器翻译的单调无限回看注意力。载于*ACL*。

Dzmitry Bahdanau、KyunghyunCho 和 Yoshua Bengio。2014。通过联合学习对齐与翻译实现神经机器翻译。*ICLR*，abs/1409.0473。

A. Berard、O. Pietquin、C. Servan 和 L. Besacier。2016 年。聆听与翻译：端到端语音到文本翻译的概念验证。载于*NIPS*。

WilliamChan、Navdeep Jaitly、Quoc V.Le 和 Oriol Vinyals。2015 年。《倾听、关注与拼写》。*ArXiv*，abs/1508.01211。

Chih-Chiang Chang 和 Hung yi Lee。2022。探索用于自适应同时语音翻译的连续积分与触发。载于*Interspeech*。

Xie Chen、Yu Wu、Zhenghao Wang、Shujie Liu 和 Jinyu Li。2020年。开发用于大规模数据集语音识别的实时流式处理 Transformer 转导器。*ICASSP*，第 5904–5908 页。

钟成成、魏瀚、张宇、潘若明、谢尔盖·基什琴科、帕特里克·阮、阿伦

Narayanan、Hank Liao、Shuyuan Zhang、Anjuli Kannan、Rohit Prabhavalkar、Z. Chen、Tara N. Sainath 和 Yonghui Wu。2019年。长篇语音识别端到端模型比较。*ASRU*。

Chung-Cheng Chiu 和 Colin Raffel。2017年。单调块式注意力。*ICLR*

Kyunghyun Cho 和 Masha Esipova。2016。神经机器翻译能否进行同步翻译？*ArXiv*，abs/1606.02012。

J. Chorowski、D. Bahdanau、D. Serdyuk、K. Cho 和 Y. Bengio。2015。基于注意力机制的语音识别模型。发表于*NIPS*。

Shun-Po Chuang、Yung-Sung Chuang、Chih-Chiang Chang 和 Hung yi Lee。2021年。研究基于连接时序分类的非自回归端到端语音翻译中的重排序能力。*ACL* 论文集，abs/2105.04840。Linhao Dong 和 Bo Xu。2020年。CIF：用于端到端语音识别的连续积分触发机制。*ICASSP 2020——2020*年 *IEEE* 国际声学、语音与信号处理会议（*ICASSP*），第 6079–6083 页。Mattia Antonino Di Gangi、Roldano Cattoni、Luisa Bentivogli、Matteo Negri 和 Marco Turchi。2019年。MuST-C：多语种语音翻译语料库。发表于 *NAACL-HLT*。Alex Graves。2012。使用循环神经网络进行序列转导。*ArXiv*，abs/1211.3711。Alex Graves、Santiago Fernández、Faustino J. Gomez 和 Jürgen Schmidhuber。2006年。连接主义时序分类：使用循环神经网络为未分段的序列数据标注。第 *23* 届国际机器学习会议论文集。Anmol Gulati、James Qin、Chung-Cheng Chiu、Niki Parmar、Yu Zhang、Jiahui Yu、Wei Han、Shibo Wang、Zhengdong Zhang、Yonghui Wu 和 Ruoming Pang。2020年。Conformer：用于语音识别的卷积增强型 Transformer。*Interspeech*，abs/2005.08100。H. Inaguma、S. Kiyono、K. Duh、S. Karita、N. Soplin、T. Hayashi 和 S. Watanabe。2020年。ESPnet-ST：一体化语音翻译工具包。发表于*ACL*。Yoon Kim 和 Alexander M. Rush。2016年。序列级知识蒸馏。发表于自然语言处理实证方法会议。T. Kudo 和 J. Richardson。2018年。SentencePiece：一种用于神经文本处理的简单且语言无关的子词分词与去分词工具。发表于 *EMNLP*。Jinyu Li。2021年。端到端自动语音识别的最新进展。*APSIPA* 信号与信息处理汇刊，abs/2111.01690。

Xian Li、Changhan Wang、Yun Tang、C. Tran、Yuqing Tang、Juan Miguel Pino、Alexei Baevski、Alexis Conneau 和 Michael Auli。2021年。通过对预训练模型进行高效微调实现多语种语音翻译。载于*ACL/IJCNLP*。Dan Liu、Mengge Du、Xiaoxi Li、Ya Li 和 Enhong Chen。2021年。用于同步翻译的交叉注意力增强转导器网络。载于*EMNLP*。Liyuan Liu、Haoming Jiang、Pengcheng He、Weizhu Chen、Xiaodong Liu、Jianfeng Gao 和 Jiawei Han。2020年。自适应学习率的方差及其他问题。*ICLR*，abs/1908.03265。Mingbo Ma、Liang Huang、Hao Xiong、Renjie Zheng、Kaibo Liu、Baigong Zheng、Chuanqiang Zhang、Zhongjun He、Hairong Liu、Xing Li、Hua Wu 和 Haifeng Wang。2019年。STACL：利用隐式预期和可控延迟，通过前缀到前缀框架实现同步翻译。载于*ACL*。Xutai Ma、Mohammad Javad Dousti、Changhan Wang、Jiatao Gu 和 Juan Pino。2020a年。SIMULEVAL：同步翻译评估工具包。载于*2020*年自然语言处理实证方法会议论文集：系统演示。计算语言学协会。Xutai Ma、Juan Pino、James Cross、Liezl Puzon 和 Jiatao Gu。2020b。单调多头注意力。载于*ICLR*。Niko Moritz、Takaaki Hori 和 Jonathan Le Roux。2019年。用于端到端语音识别的触发式注意力。*ICASSP*。Takafumi Moriya、Tomohiro Tanaka、Takanori Ashihara、Tsubasa Ochiai、Hiroshi Sato、Atsushi Ando、Ryo Masumura、Marc Delcroix 和 Taichi Asami。2021年。混合RNN-T/注意力架构的流式端到端语音识别。载于*Interspeech*。Vassil Panayotov、Guoguo Chen、D. Povey 和 S. Khudanpur。2015年。Librispeech：基于公版有声读物的自动语音识别语料库。*ICASSP*。Sara Papi、Marco Gaido、Matteo Negri 和 Marco Turchi。2022。过度生成不应得到奖励：同步语音翻译的长度自适应平均滞后。*ArXiv*，abs/2206.05807。D. Park、W. Chan、Y. Zhang、C. Chiu、B. Zoph、E. Cubuk 和 Q. Le。2019年。SpecAugment：一种用于自动语音识别的简单数据增强方法。*Interspeech*。Matt Post。2018年。呼吁明确报告BLEU分数。载于第三届机器翻译会议论文集：研究论文。Rohit Prabhavalkar、Kanishka Rao、Tara N. Sainath、Bo Li、Leif M. Johnson 和 Navdeep Jaitly。2017年。序列到序列模型在语音识别中的比较。载于*Interspeech*。

ColinRaffel、Minh-Thang Luong、Peter J. Liu、Ron J. Weiss 和 Douglas Eck。2017年。通过强制单调对齐实现在线线性时间注意力。载于*ICML*。

Tara N. Sainath、Ruoming Pang、David Rybach、Yanzhang He、Rohit Prabhavalkar、Wei Li、Mirkó Visontai、Qiao Liang、Trevor Strohman、Yonghui Wu、Ian McGraw 和 Chung-Cheng Chiu。2019年。双遍端到端语音识别。发表于*Interspeech*。

Peter Shaw、Jakob Uszkoreit 和 Ashish Vaswani。2018年。带相对位置表示的自注意力。发表于 北美计算语言学协会分会。

Yangyang Shi、Yongqiang Wang、Chunyang Wu、Ching feng Yeh、Julian Chan、Frank Zhang、Duc Le 和 Michael L. Seltzer。2020年。Emformer：一种基于高效记忆 Transformer 的声学模型，用于低延迟流式语音识别。*ICASSP*。K. Takeda、Atsunori Ogawa 和 Fumitada Itakura。1998年。通过最优词插入惩罚估算语言熵。*ICSLP*。Yun Tang、Hongyu Gong、Ning Dong、Changhan Wang、Wei-Ning Hsu、Jiatao Gu、Alexei Baevski、Xian Li、Abdelrahman Mohamed、Michael Auli 和 Juan Miguel Pino。2022。用于语音翻译和识别的统一语音-文本预训练。发表于计算语言学协会年会。Yun Tang、Juan Miguel Pino、Changhan Wang、Xutai Ma 和 Dmitriy Genzel。2021年。一种利用文本数据处理语音到文本任务的通用多任务学习框架。*ICASSP*。Ashish Vaswani、Noam M. Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N. Gomez、Lukasz Kaiser 和 Illia Polosukhin。2017年。注意力就是一切。*NIPS*，abs/1706.03762。Changhan Wang、Yun Tang、Xutai Ma、Anne Wu、Dmytro Okhonko 和 Juan Miguel Pino。2020年。Fairseq S2T：使用 fairseq 进行快速语音到文本建模。发表于*AACL*。Peidong Wang、Eric Sun、Jian Xue、Yu Wu、Long Zhou、Yashesh Gaur、Shujie Liu 和 Jinyu Li。2022。LA MASSU：使用神经转导器进行流式、语言无关的多语言语音识别与翻译。*ArXiv*，abs/2211.02809。Shinji Watanabe、Takaaki Hori、Suyoun Kim、John R. Hershey 和 Tomoki Hayashi。2017年。用于端到端语音识别的混合 CTC/注意力架构。*IEEE* 信号处理精选主题期刊，11:1240–1253。R. Weiss、J. Chorowski、N. Jaitly、Y. Wu 和 Z. Chen。2017年。序列到序列模型可以直接翻译外语语音。发表于*Interspeech*。

Chunyang Wu、Yongqiang Wang、Yangyang Shi、Ching feng Yeh 和 Frank Zhang。2020年。使用带增强记忆的自注意力，实现基于 Transformer 的流式声学模型。发表于*Interspeech*。

Jian Xue、Peidong Wang、Jinyu Li、Matt Post 和 YasheshGaur。2022。基于神经转导器的大规模流式端到端语音翻译。发表于*Interspeech*。

Brian Yan、Siddharth Dalmia、Yosuke Higuchi、Graham Neubig、Florian Metze、Alan W. Black 和 Shinji Watanabe。2022。连接时序分类对齐可改进自回归翻译。*ArXiv*，abs/2210.05200。

Qian Zhang、Han Lu、Hasim Sak、Anshuman Tripathi、Erik McDermott、Stephen Koo 和 Shankar Kumar。2020年。《Transformer 转导器：一种采用 Transformer 编码器和 RNN-T 损失的可流式语音识别模型》。*ICASSP*，第 7829–7833 页。

MUST-C	小时	#W(m)
EN-DE	408	4.2
EN-ES	504	5.2

表 7：MUST-C 训练数据集的数据统计。“#W(m)”表示“词数（百万）”。

CS（毫秒）	WER	AL	LAAL	AP	DAL
160	16.20	809	856	0.63	1177
320	13.79	868	914	0.65	1254
480	13.68	1063	1102	0.70	1485
640	13.14	1317	1350	0.71	1755

表 8：不同块大小下转导器的词错误率与延迟。

A MUST-C 数据集统计

我们在 MuST-C（Gangi 等人，2019年）上开展实验。自动语音识别实验使用 EN→ES 方向的数据英语部分。语音翻译实验涵盖两个翻译方向：EN→ES 和 EN→DE。详细的训练数据统计信息见表 7。第二列是语音训练数据的总小时数。第三列是（源）词数。

B 流式处理详细结果

本节介绍详细的流式处理实验结果。我们报告了 SimulEval 工具包提供的不同延迟指标（ Ma 等人，2020年a），包括平均滞后（AL）（Ma 等人，2019年）、平均比例（AP）（Cho 和 Esipova，2016）、可微平均滞后（DAL）（ Arivazhagan 等人，2019年）以及长度自适应平均滞后（LAAL）（Papi 等人，2022）。AL、DAL 和 LAAL 的单位均为毫秒。我们报告了不同块大小下的评估结果，块大小分别为 160、320、480 和 640 毫秒，见表8 至表 13。“CS”在这些表中表示块大小。流式自动语音识别结果以词错误率（WER）表示（表 8和表9）。表 10、表 11、表 12 和表 13中报告了两个翻译方向的 BLEU 分数。

C Librispeech 自动语音识别结果

模型配置与**MuST-C** 第 §5.1节中的实验相同。模型使用 16 个 A100 GPU 进行训练，每个 GPU 的批次大小为 20k 个语音帧，共训练 300k 次更新。SpecAugment（Park

CS（毫秒）	词错误率	AL	LAAL	AP	DAL
160	13.97	787	829	0.63	1146
320	12.71	907	947	0.65	1284
480	11.75	1061	1097	0.69	1482
640	11.59	1293	1323	0.74	1742

表9：不同块大小下 TAED 词错误率与延迟的关系（$\lambda = 1.4$）。

CS（毫秒）	BLEU	AL	LAAL	AP	DAL
160	25.7	1170	1379	0.70	1669
320	26.7	1137	1357	0.73	1679
480	27.3	1205	1433	0.74	1784
640	27.7	1356	1578	0.77	1951

表10：不同块大小下 Transducer BLEU 与延迟的关系（EN→ES）。

CS（毫秒）	BLEU	AL	LAAL	AP	DAL
160	26.19	1000	1224	0.71	1623
320	27.61	1120	1351	0.74	1795
480	27.50	1244	1477	0.77	1930
640	28.35	1473	1693	0.79	2188

表11：不同分块大小下 TAED 的 BLEU 与延迟对比（EN→ES）（$\lambda = 1.4$）。

CS（毫秒）	BLEU	AL	LAAL	AP	DAL
160	20.76	1282	1412	0.68	1618
320	21.80	1252	1389	0.70	1612
480	22.52	1306	1447	0.72	1717
640	23.32	1498	1630	0.75	1925

表12：不同分块大小下转导器的 BLEU 与延迟对比（EN→DE）。

CS（毫秒）	BLEU	AL	LAAL	AP	DAL
160	21.57	1263	1411	0.72	1823
320	22.63	1354	1530	0.74	2007
480	23.48	1369	1554	0.77	2088
640	23.47	1903	2024	0.82	2597

表13：不同分块大小下 TAED 的 BLEU 与延迟对比（EN→DE）（$\lambda = 1.2$）。

模型	test		dev	
	干净	其他	干净	其他
转导器（ofl）	3.2	8.0	3.0	8.2
转导器（流式处理）	4.4	11.2	4.0	11.3
TAED（离线）	3.1	7.4	2.9	7.4
TAED（流式处理）	4.2	10.4	4.1	10.8

表 14：Librispeech 数据集上离线和流式处理自动语音识别的比较。“ofl”和“str”分别代表离线模型和流式处理模型。

等人，2019年）不使用时间扭曲，dropout 设置为 0.1。我们每隔 2500 次更新保存检查点，并对最优的 10 个检查点进行平均，用于基于贪心搜索的推理。模型在包含 960 小时语音的 **Librispeech** （Panay- otov 等人，2015年）训练集上进行训练，并在 4 个测试集/开发集上进行评估。在表 14中，流式模型（“str”）使用块大小为 320 毫秒、带有一个右侧前瞻块的设置进行训练。TAED 在两个干净（简单）数据集上取得了相近的词错误率，并在另外两个（困难）数据集上将词错误率降低了 0.5 至 0.8。