基于 Transformer 的移动设备统一关键词检测与音频标注

*He*inrich Dinkel†、*Yongqing Wang*†、*Zhiyong Yan*†、*Junbo Zhang* 和 *Yujun Wan* *g*

小米公司，中国北京

arXiv:2303.01812v1 [cs.SD] 3 Mar 2023

摘要

关键词检出（KWS）是大多数现代智能助手中一项核心的人机交互前端任务。近年来，有人提出了一种统一框架（UniKW-AT），为关键词识别模型增添了音频标注（AT）能力。然而，以往的研究未考虑 UniKW-AT 模型的实际部署，而在实际部署中，模型大小和推理速度等因素比单纯的性能更为重要。本文介绍了三种可部署于移动设备的模型，统称为统一 Transformer（UiT）。其中，表现最好的模型在 Audioset 上的平均精度均值（mAP）达到 34.09，在公开的 Google Speech Commands V1 数据集上取得了 97.76 的准确率。此外，我们在四种移动平台上对提出的方法进行了基准测试，结果表明，提出的 UiT 模型相较于具有竞争力的 MobileNetV2，速度可提升 2 至 6 倍。

索引词— 关键词检出、音频标注、视觉 Transformer、弱监督学习。

1. 引言

关键词检出（KWS）是目前大多数智能语音助手的一项关键前端任务：当用户说出特定关键词短语时，它会触发与语音助手的交互。此外，音频标注（AT）旨在将特定音频内容标注为声音事件类别，例如婴儿啼哭声。在先前的工作中， [1], 作者已证明，通过统一框架（UniKW-AT）对这两项任务进行建模是可行的，并且在不牺牲 KWS 准确率的情况下，显著提升了噪声鲁棒性。

然而，若将 UniKW-AT 模型部署于真实场景中，它们就需要满足与关键词识别模型相同的要求。首先，关键词识别模型运行于端侧，其规模（即参数数量）受到限制。其次，由于需要“始终运行”，关键词识别模型必须具备较快的推理速度，并尽可能减少浮点运算（FLOPs）。第三，关键词识别模型的延迟需要尽可能低，以便助手能够立即唤醒。尽管关键词识别领域已经对这些要求进行了充分研究，但据我们所知，此前尚无研究关注音频标注模型的轻量化与端侧计算。本研究旨在弥补这一空白，并介绍多种用于统一关键词识别与音频标注（UniKW-AT）的基于 Transformer 的模型，以满足上述要求。

这种 UniKW-AT 模型的优势在于，音频标注分支的输出可以进一步传递到自动语音识别（ASR）流水线中，从而可能增强噪声鲁棒性。至少，这种 UniKW-AT 模型可以充当语音活动检测器 [2, 3, 4]。

1.1. 先前工作

鉴于关键词识别具有重要的实际意义，先前的工作主要致力于减小模型的参数规模 [5], 、提高其推理速度 [6] 以及降低误接受率 [7]。在架构方面，卷积神经网络（CNNs）已得到该领域的广泛研究 [8, 9, 5], ；近年来，基于 Transformer 的模型 [10, 11, 12, 13] 和多层感知机（MLP）混合器 [14, 15] 也得到了研究。相比之下，在音频标注领域，大多数工作都着重于提升模型在广为人知的 Audioset 基准上的当前最佳性能。 [16]等工作推动了卷积神经网络的发展，而 [17] 则采用了 Transformer。然而，音频标注领域的大多数研究仅关注性能提升，而未考虑这些模型的实际部署。

2. 统一的关键词识别与音频标注 Transformer

本文提出了多种基于 Transformer 的网络，以下简称统一 *T*ransformer（UiT），旨在保持关键词识别和音频标注性能的同时，提高推理速度并减小模型参数规模和计算开销。

统一关键词识别与音频标注UniKW- AT 已在 [1]中提出，其建模方式如下。给定包含 K 个关键词的目标 KWS 标签集 $\mathbb{L}_{\text{KWS}}$，以及包含 C 个声音事件的音频标注标签集 $\mathbb{L}_{\text{AT}}$，UniKW-AT 将二者合并

† 贡献相同。

得到 $\mathbb{L} = \mathbb{L}_{\text{KWS}} \cup \mathbb{L}_{\text{AT}}$。从 KWS 和音频标注数据集中采样的训练样本会随机裁剪至某个目标时长 t，并通过二元交叉熵（BCE）损失对该框架进行优化。整个训练框架如图1所示。

视觉 Transformer Transformer最初于 [18] 被提出用于机器翻译，并迅速成为自然语言处理（NLP）领域的最先进方法。随后，在 [19], , 视觉变换器（ViT）被提出，作为将Transformer应用于计算机视觉领域的一种改编。之后，基于ViT的Transformer被用于音频标注，其中图像被二维频谱图取代 [17, 20]。ViT框架的核心思想是“分块化”操作，即首先将输入图像（此处为频谱图）拆分为 N 个互不重叠的块。每个块的大小为 $P = P_{\text{T}} \times P_{\text{F}}$（时间和频率），并通过卷积操作提取。随后，这些块被输入由 L 个相同模块组成的Transformer模型，每个模块包含一个多头注意力（MHA）层，后接一个多层感知机（MLP）层。
MHA层计算：

$$\mathbf{A} = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D}})\mathbf{V}, \tag{1}$$

其中，$\mathbf{K} \in \mathbb{R}^{N\times D}$,$\mathbf{Q} \in \mathbb{R}^{N\times D}$,$\mathbf{V} \in \mathbb{R}^{N\times D}$ 分别是键、查询和值矩阵，由输入$\mathbf{X} \in \mathbb{R}^{N\times D}$经线性变换$\mathbf{W}_j \in \mathbb{R}^{D\times D}$, $j \in \{K, Q, V\}$得到。单个模块的复杂度为 $\mathcal{O}(N^2D + D^2N)$，即对于模型维度 D 和块数 N呈二次增长。

2.1. 提出的方法

我们发现，嵌入维度 D 和块数 N 是导致计算开销过大的主要原因，因此提出采用分块缩减和瓶颈注意力机制来降低我们的UiT模型的复杂度。

分块缩减与下采样 Audioset 上常见的 Transformer 模型以 10 秒为时间尺度进行训练，使用多达 $N = N_{\text{T}} \times N_{\text{F}} = 1212$ [20, 17] 个块，导致 Transformer 模型面临较高的内存和计算负担。由于时频信息对音频标注至关重要，我们着重减少可用时间分块的数量N_{T}，并通过向模型输入长度为 t （即 1 秒）的裁剪片段，将频率分块数 N_{F} 固定。限制上下文会损害模型在 Audioset 上的音频标注性能，因为模型的训练时长（1 秒）与评估时长（10 秒）不匹配。不过，伪强标签（PSL） [21] 可以通过利用预训练教师模型（MobileNetV2）预测精细尺度的标签，部分缓解性能下降。

大多数CNN模型中的第一层通常称为词干，将 $1 \times T \times F$ 频谱图输入映射到

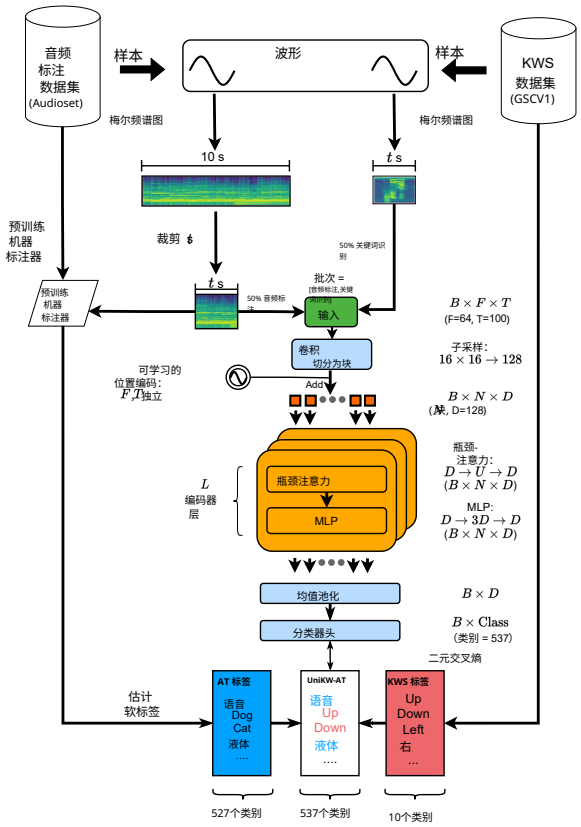


图 1。用于训练的 UiT 框架示意图。我们从 AT（Audioset）数据集和 KWS 数据集（GSCV1）中采样。然后，我们随机裁剪 Audioset 样本，使其长度与 KWS 目标样本匹配 t。随后，使用预训练模型（MobileNetV2）为这些 Audioset 样本估计标签（伪强标签）。训练期间，每个批次由 50% 的 AT 数据和 50% 的 KWS 数据组成。训练通过优化二元交叉熵（BCE）准则完成。

$C \times \frac{T}{2} \times \frac{F}{2}$, 其中 $C \geq 16$ [22]，这意味着所需的总内存将扩大 $\geq$ 4倍。我们的方法使用下采样词干，通过将大小为 P的输入补丁映射到低维空间 D来直接降低内存需求，其中 $D < P$。

瓶颈注意力 瓶颈注意力（BN-A）着重于在自注意力阶段降低维度 D，如公式（1）所示。我们的直觉是，频谱图中的每个 patch 嵌入都包含大量冗余信息。因此，我们提出采用 BN-A 方法，在自注意力过程中将维度 D 降至更低的空间 $U, U < D$。我们为所有模型设定 $U = \frac{D}{4}$。

架构 提出的模型架构如表 1 所示。所有架构均使用 $P = 256 =$

模型	L	D	MLP	#参数量	百万浮点运算次数	M_{pk}
UiT-XS	12	128	384	1.5 M	34	7.59
UiT-2XS	6	128	384	0.8 M	18	4.10
UiT-3XS	4	128	384	574 k	13	3.15

表 1。基于 UiT 的提出的模型架构。MFLOPs 数量和峰值内存使用量 M_{pk}（以 MB 为单位）是按 1 s 计算的。

16×16 图块尺寸，使模型延迟为 16 帧，并且每个自注意力操作使用两个注意力头。因此，沿频率轴有 $N_{\mathrm{F}} = 4$ 个块，每秒输入音频有 $N_{\mathrm{T}} = 6$ 个块。由于 ReLU 的推理速度更快且计算复杂度更低，我们将其用作默认激活函数。与 [11], 类似，我们在多层感知机中使用较低的嵌入维度 $3D$，从而减少内存占用。

3. 实验

3.1. 数据集

本文主要使用 Google语音命令数据集V1（GSCV1） [23] 和 Audioset [24]数据集。我们使用 GSCV1 的常见 11类子集（V1-11），其中原有的 30 个类别缩减为 10 个常见关键词：“Yes”“No”“向上”“向下”“Left”“Right”“On”“Off”“Stop”“Go”，其余 20 个关键词则标记为 AS 标签“语音”，每个样本时长为 1 秒。我们使用官方的训练/验证/测试划分，分别包含 51,088/6,798/6,835 个语音片段。对于音频标注训练，我们使用 AS 中时长为 60 小时的平衡子集，其中包含 21,292 个音频片段，每个片段的时长不超过 10 秒。

评估 评估分为关键词识别和音频标注子任务。音频标注评估使用 AS 的常见评估子集，其中包含 18,229 个音频片段，总时长为 50 小时。关键词识别分析主要针对 GSCV1 数据集，该数据集提供了 2,567 个目标关键词和 4,268 个非目标样本。请注意，对于时长超过目标长度 t（即 10 秒）的片段，我们会将输入拆分为长度为 t（即 1 秒）的分块，然后将这些分块输入模型，并对所有输出分数取平均。

3.2. 设置

关于前端特征提取，我们使用对数梅尔频谱图（LMS），每 10 毫秒提取一次，采用 64 个频带、32 毫秒的窗口和 16 kHz 的采样率。我们的 UiT 变换器模型使用与时间和频率无关的可学习位置嵌入。在波形域中，我们采用随机移位、音量增益和极性反转作为数据增强方法，获得频谱图后，我们使用 SpecAugment [25]对数据进行增强。训练采用批量大小为 64，最多训练 800 个训练轮次，并使用 AdamW 优化 [26] ；先以 20 个训练轮次进行线性预热，初始学习率为 0.001，随后通过余弦退火逐步降低学习率。训练期间，我们使用平均精度均值（mAP）作为主要验证指标。由于模型可以同时预测多个标签，因此计算 UniKW-AT 的关键词识别准确率需要进行后处理，即“关键词 +语音”。在这里，我们使用 $\gamma = 0.2$ 作为阈值，用于判断关键词是否存在。在联合留出验证数据集（ GSCV1-valid 和 AS-balanced）上取得最高 mAP 的前 4 个模型将提交进行评估。神经网络后端使用 PyTorch [27] 实现。为加快训练过程，我们使用掩码自编码器方法 [20] 在完整的 Audioset 上预训练单个 UiT-XS，并从该单一检查点初始化每个模型的所有相关层。源代码已公开发布[1]。

4. 实验

4.1. 主要结果

我们在 GSCV1 和 Audioset 上的核心结果见表 2。为了说明训练 UniKW-AT 模型的难度，我们还使用常见的 TC-ResNet8 [5]模型进行了基线实验。结果表明，尽管 TC-ResNet8 在 GSCV1 上表现出色（准确率为 96.72），但在音频标注（AT）上却无法取得有意义的性能（平均精度均值为 8.67）。值得注意的是，由于 UniKW-AT 的训练中加入了来自 Audioset 的 60 小时“噪声”样本，增强了模型对误报的鲁棒性，因此 TC-ResNet8 在 GSCV1 上的表现优于公开报告的结果（$96.10 \rightarrow 96.72$）。如我们所见，与先前基于 MobileNetV2（MBv2）的 UniKW-AT 方法，以及文献中其他关于关键词识别和音频标注性能的研究相比，我们提出的 UiT-XS 取得了具有竞争力的结果。

4.2. 移动硬件上的推理延迟

本文使用 PyTorch 移动端基准测试工具[2]测量模型的推理速度。表 3 展示了在四种不同移动设备上测得的推理速度：我们使用两款高端高通骁龙芯片——骁龙 865（SD865）和骁龙 888（SD888），以及两款中端芯片——联发科 Helio G90T（G90T）和联发科天玑 700（MT700）。我们将结果与 TC-ResNet8 和 MBv2 进行比较。MBv2 可视为基线，代表以往的 UniKW-AT 方法，而

[1] www.github.com/Richermans/UIT_移动端[2] https://pytorch.org/tutorials/recipes/mobile_perf.html

方法	#参数量（M）	GSCV1	AS
TC-ResNet8 [5]	0.06	96.10	-
NAS2 [6]	0.88	97.22	-
MEGA [28]	0.3	96.92	-
MatchBoxNet [29]	0.5	96.83	-
KWT-1 [10]	0.6	97.05	-
LETR-128 [11]	0.6	97.61	-
LETR-256 [11]	1.1	97.85	-
KWT-2 [10]	2.4	97.36	-
KWT-3 [10]	5.3	97.24	-
Wav2KWS [30]	225	97.90	-
MBv2 [31]	2.9	-	26.50
Eff-B0 [31]	5.3	-	33.50
Eff-B2 [31]	13.6	-	34.06
ResNet-50 [31]	25.6	-	31.80
CNN14 [16]	76	-	27.80
AudioMAE [20]	80	-	37.10
TC-ResNet8	0.1	96.72	8.67
MBv2 [1]	2.9	97.53	33.42
MBv2◆ [1]	2.9	97.53	32.51
UiT-XS◆	1.5	97.76	34.09
UiT-2XS◆	0.8	97.31	32.21
UiT-3XS◆	0.6	97.18	30.97

表 2。我们提出的 UiT 方法与文献中其他工作的比较。GSCV1 的结果采用准确率，AS 的结果采用平均精度均值。标有 ◆ 的方法在 1 秒分块上进行评估，这会影响 Audioset 的表现。结果中的“-”表示暂无数据。

TC-ResNet8 代表现代关键词识别系统对速度的要求。

延迟结果表明，使用 UiT-XS 时，我们提出的方法相较于 MBv2 可实现超过 2 倍的加速（8.0 → 3.4 毫秒），同时性能相近（见表 2）。虽然 UiT-2XS 和 UiT-3XS 的速度慢于基线 TC-ResNet8，但它们在音频标注方面表现出色（32.21/30.97 对比 8.67 平均精度均值，见表 2）。另一个值得注意的重要因素是，基线 MBv2 的延迟为 320 毫秒，而我们提出的模型可在 160 毫秒内响应。

4.3. 消融

这里我们针对提出的 BN-A 机制，以及选择 ReLU 激活函数而非更常见的高斯误差线性单元（GeLU）[32]进行了消融实验。为简单起见，这些测试仅使用 SD865 芯片组。结果见表4。我们观察到，与标准自注意力相比，提出的 BN-A 机制可将推理时间至少缩短 20%（4.1 → 3.4 毫秒），且性能没有明显差异。此外，尽管采用 GeLU 的标准自注意力相比提出的

模型	SD865	SD888	G90T	MT700
TC-ResNet8	0.4	0.4	1.1	1.1
MBv2	8.0	6.2	13.1	11.6
UiT-XS	3.4	3.4	7.3	7.1
UiT-2XS	1.7	1.5	2.8	3.2
UiT-3XS	1.2	1.1	2.2	2.2

表 3. 在移动系统级芯片（SoC）平台上的推理速度比较，在中央处理器（CPU）上以 float32 精度运行，并以毫秒计量。每次速度评估均先通过十次预热迭代预热芯片，随后对长度为 1 秒的输入进行 1000 次测试试验。

BN-A + ReLU 方法能带来些许性能提升，但也会显著降低推理速度（UiT-XS 为 3.4 → 5.7 毫秒），并增加峰值内存使用量（UiT-XS 为 7.59 → 10.28 MB），从而限制其在现实场景中的应用潜力。

模型	消融	GSCV1	AS	速度	M_{pk}
UiT-XS	提出的方法	**97.76**	34.09	**3.4**	**7.59**
	不使用 BN-A	97.75	33.76	4.1	10.28
	ReLU →GeLU	97.69	**34.12**	5.7	10.28
UiT-2XS	提出的方法	**97.31**	32.21	**1.7**	**4.10**
	不含 BN-A	96.94	32.30	2.2	5.44
	ReLU →GeLU	97.27	**32.41**	2.9	5.44
UiT-3XS	提出的方法	**97.18**	30.97	**1.2**	**3.15**
	不含 BN-A	96.71	30.87	1.6	3.90
	ReLU →GeLU	96.91	**30.99**	2.1	3.90

表4。提出的模型的消融实验，其中“速度”表示测得的毫秒数，M_{pk} 是峰值内存需求（MB）。“w/o BN-A”表示采用标准自注意力机制，而“ReLU → GeLU”则采用 GeLU 和标准自注意力。结果是在 1 秒的输入上评估的。最佳结果以粗体标示。

5. 结论

本文提出了基于视觉变换器的Transformer模型（UiT），针对 UniKW-AT 在移动设备上的部署进行了优化。我们介绍了三种轻量化 UiT 模型，它们在 UniKW-AT 上表现出色，同时具备快速推理速度、较低的峰值内存使用量和较小的参数量。我们的最佳模型（UiT-XS）在 GSCV1 数据集上达到 97.76 的准确率，在 Audioset 上达到 34.09 的平均精度均值，性能优于有竞争力的 MBv2 基线，同时参数量仅为其一半，推理速度也大幅提升。

6. 参考文献

[1] Heinrich Dinkel、Yongqing Wang、Zhiyong Yan、Junbo Zhang 和 Yu-jun Wang，《UniKW-AT：统一关键词检出与音频标注》，发表于*Interspeech 2022* 论文集，2022年，第 3238–3242 页。[2] Heinrich Dinkel、Shuai Wang、Xuenan Xu、Mengyue Wu 和 Kai Yu，《真实环境中的语音活动检测：一种基于数据驱动的教师-学生训练方法》，《*IEEE/ACM*音频、语音与语言处理汇刊》，第 29 卷，第 1542–1555 页，2021年。[3] Xuenan Xu、Heinrich Dinkel、Mengyue Wu 和 Kai Yu，《一种用于真实环境在线语音活动检测的轻量化框架》，发表于 *Interspeech*，2021年，第 371–375 页。[4] Sashi Novitasari、Takashi Fukuda 和 Gakuto Kurata，《通过语音活动检测集成提升噪声环境下自动语音识别的鲁棒性》，发表于*Interspeech 2022* 论文集，2022年，第 3784–3788 页。[5] Seungwoo Choi、Seokjun Seo、Beomjun Shin、Hyeongmin Byun、Martin Kersner、Beomsu Kim、Dongyoung Kim 和 Sungjoo Ha，《面向移动设备实时关键词检出的时序卷积》，发表于*Interspeech 2019* 论文集，2019年，第 3372–3376 页。[6] Tong Mo、Yakun Yu、Mohammad Salameh、Di Niu 和 Shangling Jui，《用于关键词检出的神经架构搜索》，发表于*Interspeech 2020* 论文集，2020年，第 1982–1986 页。[7] Anirudh Raju、Sankaran Panchapagesan、Xing Liu、Arindam Mandal 和 Nikko Strom，《用于提升回放干扰下关键词检出鲁棒性的数据增强》，《*arXiv* 预印本 *arXiv:1808.00563*》，2018年。[8] Tara N. Sainath 和 Carolina Parada，《用于小占用关键词检出的卷积神经网络》，发表于 *Interspeech 2015* 论文集，2015年，第 1478–1482 页。[9] Byeonggeun Kim、Simyung Chang、Jinkyu Lee 和 Dooyong Sung，《用于高效关键词检出的广播残差学习》，发表于*Interspeech 2021* 论文集，2021年，第 4538–4542 页。[10] Axel Berg、Mark O'Connor 和 Miguel Tairum Cruz，《关键词Transformer：用于关键词检出的自注意力模型》，发表于*Interspeech 2021* 论文集，2021年，第 4249–4253 页。[11] Kevin Ding、Martin Zong、Jiakui Li 和 Baoxiang Li，《Letr：一种轻量高效的关键词检出Transformer》，发表于*ICASSP 2022 - 2022 IEEE*国际声学、语音与信号处理会议（*ICASSP*），2022年，第 7987–7991 页。[12] Yiming Wang、Hang Lv、Daniel Povey、Lei Xie 和 Sanjeev Khudanpur，《使用流式Transformer进行唤醒词检测》，发表于*ICASSP 2021-2021 IEEE*国际声学、语音与信号处理会议（*ICASSP*）。IEEE，2021年，第 5864–5868 页。[13] Vineet Garg、Wonil Chang、Siddharth Sigtia、Saurabh Adya、Pramod Simha、Pranay Dighe 和 Chandra Dhir，《用于硬件高效语音触发检测及误触发抑制的流式Transformer》，发表于*Interspeech 2021* 论文集，2021年，第 4209–4213 页。[14] Mashrur M Morshed 和 Ahmad Omar Ahsan，《无注意力的关键词检出》，《*arXiv* 预印本 *arXiv:2110.07749*》，2021年。[15] Dianwen Ng、Yunqi Chen、Biao Tian、Qiang Fu 和 Eng Siong Chng，《Convmixer：结合课程学习的特征交互卷积，用于小占用及噪声远场关键词检出》，《*arXiv* 预印本 *arXiv:2201.05863*》，2022年。[16] Qiuqiang Kong、Yin Cao、Turab Iqbal、Yuxuan Wang、Wenwu Wang 和 Mark D. Plumbley，《PANNs：用于音频模式识别的大规模预训练音频神经网络》，《*IEEE/ACM*音频、语音与语言处理汇刊》，第 28 卷，第 2880–2894 页，2020年12月。[17] Yuan Gong、Yu-An Chung 和 James Glass，《AST：音频频谱图 Transformer》，发表于*Interspeech 2021* 论文集，2021年，第 571–575 页。[18] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Lukasz Kaiser 和 Illia Polosukhin，《注意力就是你所需要的一切》，《神经信息处理系统进展》，第 30 卷，2017年。

[19] Alexander Kolesnikov、Alexey Dosovitskiy、Dirk Weissenborn、Georg Heigold、Jakob Uszkoreit、Lucas Beyer、Matthias Minderer、Mostafa Dehghani、Neil Houlsby、Sylvain Gelly、Thomas Unterthiner 和 Xi- aohua Zhai，“一张图像胜过 16×16 个词：用于大规模图像识别的 Transformer”，2021年。[20] Hu Xu、Juncheng Li、Alexei Baevski、Michael Auli、Wojciech Galuba、Florian Metze、Christoph Feichtenhofer 等，“能够聆听的掩码自编码器”，*arXiv* 预印本 *arXiv:2207.06405*，2022年。[21] Heinrich Dinkel、Zhiyong Yan、Yongqing Wang、Junbo Zhang 和 Yu- jun Wang，“用于大规模弱监督音频标注的伪强标签”，载于 *ICASSP 2022 - 2022 IEEE* 国际声学、语音与信号处理会议（*ICASSP*），2022年，第336–340页。[22] Mark Sandler、Andrew Howard、Menglong Zhu、Andrey Zhmoginov 和 Liang-Chieh Chen，“MobileNetV2：倒置残差与线性瓶颈”，载于 *IEEE* 计算机视觉与模式识别会议论文集，2018年，第4510–4520页。[23] Pete Warden，“Speech Commands：一个有限词汇语音识别数据集”，*arXiv* 预印本 *arXiv:1804.03209*，2018年。[24] Jort F Gemmeke、Daniel PW Ellis、Dylan Freedman、Aren Jansen、Wade Lawrence、R Channing Moore、Manoj Plakal 和 Marvin Ritter，“AudioSet：用于音频事件的本体和人工标注数据集”，载于 *2017 IEEE* 国际声学、语音与信号处理会议（*ICASSP*）。IEEE，2017年，第776–780页。[25] Daniel S. Park、William Chan、Yu Zhang、Chung-Cheng Chiu、Barret Zoph、Ekin D. Cubuk 和 Quoc V. Le，“SpecAugment：一种用于自动语音识别的简单数据增强方法”，载于 *Interspeech 2019* 论文集，2019年，第2613–2617页。[26] Tim Dettmers、Mike Lewis、Sam Shleifer 和 Luke Zettlemoyer，“通过分块量化实现 8位优化器”，载于 国际学习表征会议，2021年。[27] Adam Paszke、Sam Gross、Francisco Massa、Adam Lerer、James Bradbury、Gregory Chanan、Trevor Killeen、Zeming Lin、Natalia Gimelshein、Luca Antiga、Alban Desmaison、Andreas Kopf、Edward Yang、Zachary DeVito、Martin Raison、Alykhan Tejani、Sasank Chil- amkurthy、Benoit Steiner、Lu Fang、Junjie Bai 和 Soumith Chintala，“PyTorch：一种命令式风格的高性能深度学习库”，载于 神经信息处理系统进展第*32*卷，H Wallach、H Larochelle、A Beygelzimer、F Alch´e-Buc、E Fox 和 R Garnett 编，第8026–8037页。Curran Associates, Inc.，2019年。[28] Xuezhe Ma、Chunting Zhou、Xiang Kong、Junxian He、Liangke Gui、Graham Neubig、Jonathan May 和 Luke Zettlemoyer，“MEGA：配备移动平均的门控注意力”，*arXiv* 预印本 *arXiv:2209.10655*，2022年。[29] Somshubra Majumdar 和 Boris Ginsburg，“MatchBoxNet：用于语音命令识别的 1D 时间通道可分离卷积神经网络架构”，载于 *Interspeech 2020* 论文集，2020年，第3356–3360页。[30] Deokjin Seo、Heung-Seon Oh 和 Yuchul Jung，“Wav2KWS：从语音表征迁移学习以实现关键词检出”，*IEEE Access*，第9卷，第80682–80691页，2021年。[31] Yuan Gong、Yu-An Chung 和 James Glass，“PSLA：通过预训练、采样、标注和聚合改进音频事件分类”，*arXiv* 预印本 *arXiv:2102.01243*，2021年。[32] Dan Hendrycks 和 Kevin Gimpel，“高斯误差线性单元（GeLU）”，*arXiv* 预印本 *arXiv:1606.08415*，2016年。