



声音感知技术的无障碍协同应用

张俊博



小米 AI 实验室 - 声学语音组



手机与手机周边

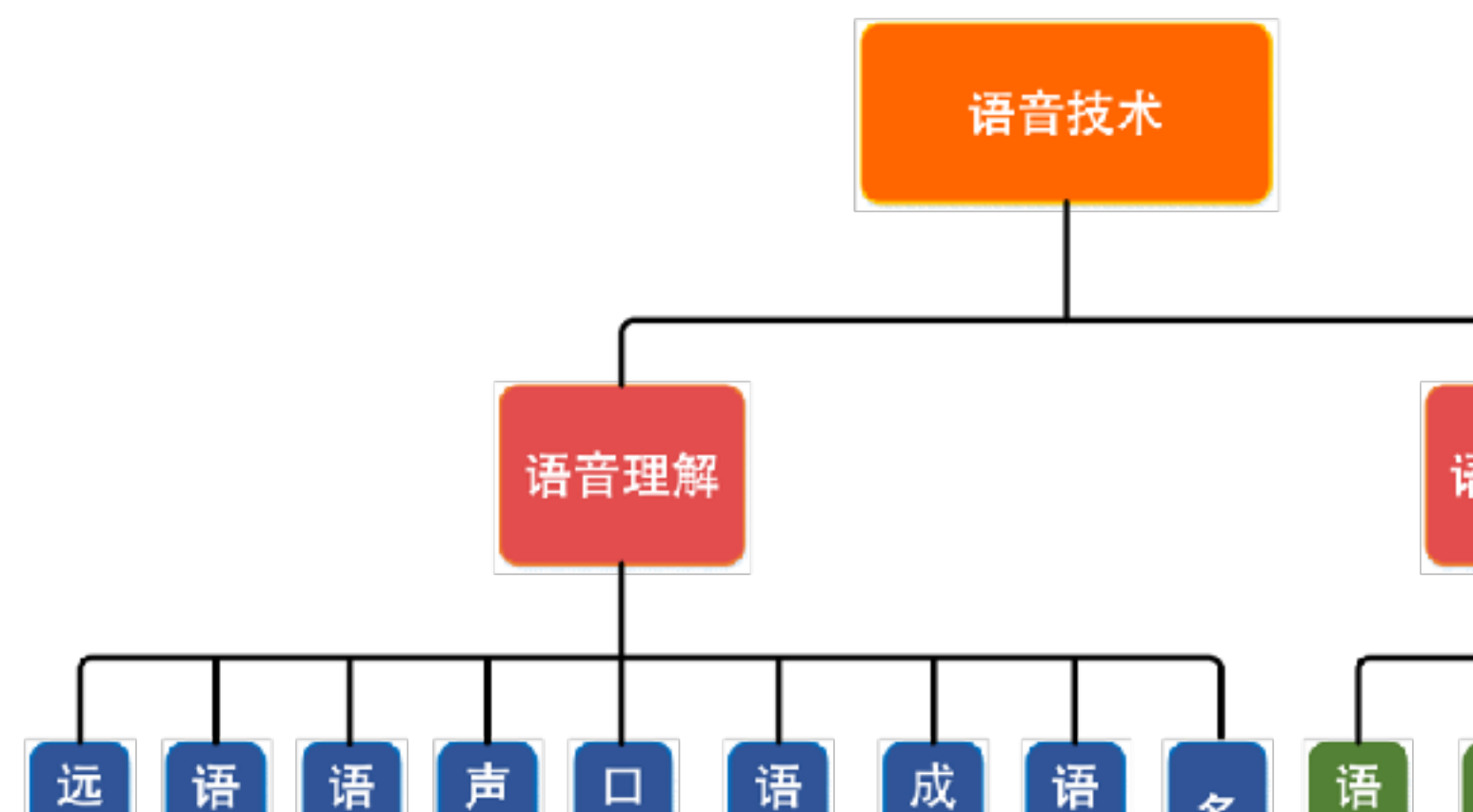
AIoT

MIUI

销服



Mi-Speech





从一届黑客马拉松比赛说起

作品 初衷

<2020> HACKATHON

我们的队伍叫DAKUI，是因为一位叫张大奎的朋友。他出生于河南焦作农村，博士毕业于北京理工大学计算机专业。因自幼患脑性瘫痪，走路、说话都比一般人更加困难。脑瘫会让肌肉协调能力受损，大奎说话比较费劲，口齿也不太清楚，和人交流的时候，听的人也需要全神贯注，配合推测和确认，才能理解，双方都挺费劲，这也阻碍了大奎和朋友们的交流。

除了大奎，还有很多人，都需要AI辅助与替代沟通技术。据统计推测，中国有600多万脑瘫患者，他们在努力地工作，认真地生活。于是我们做了这款名为“聆听”的软件，希望大家能更好地聆听脑瘫患者的声音，让交流变得更加轻松。

让全球每个人，都能享受科技带来的美好生活。



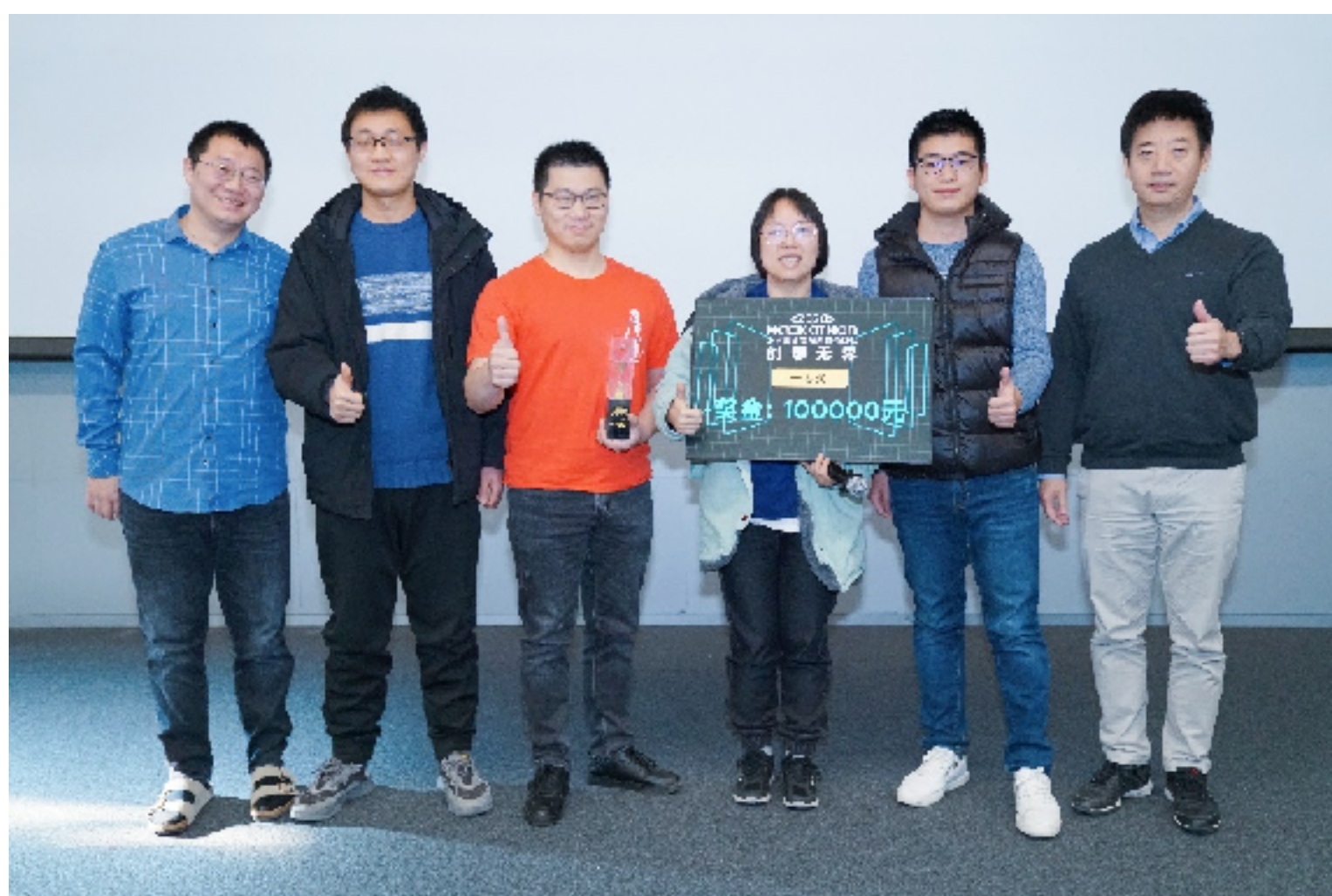
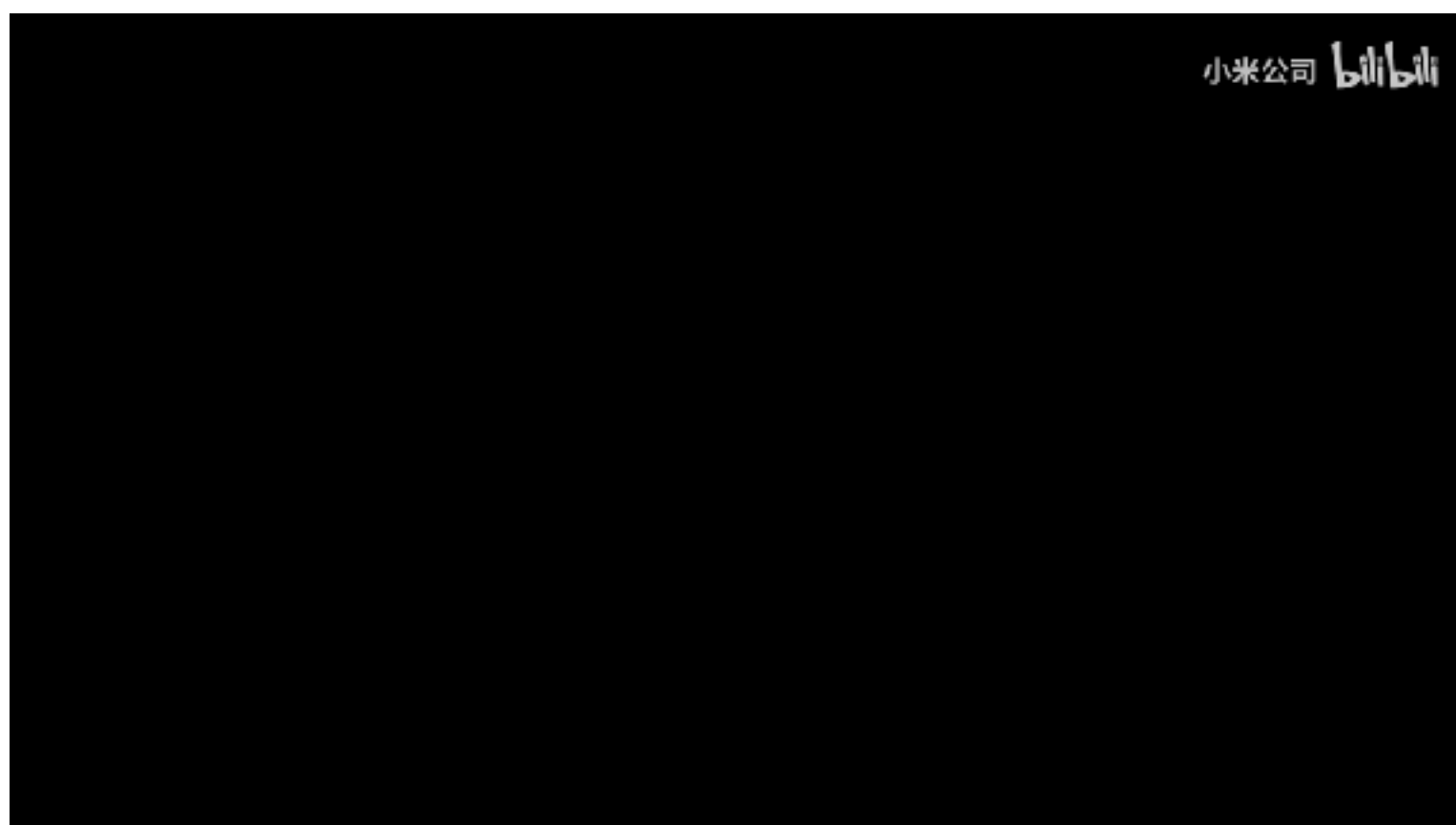
计算机博士张大奎



外卖员许龙庆



诗人余秀华



Empirical Evaluation of Speaker Adaptation on DNN based Acoustic Model

Ke Wang^{1,2}, Junbo Zhang², Yujun Wang², Lei Xie^{1*}

¹Shaanxi Provincial Key Laboratory of Speech and Image Information Processing,
School of Computer Science, Northwestern Polytechnical University, Xi'an, China
²Xiaomi, Beijing, China

{xewang, lxie}@nwpu-asip.org, {zhangjunbo, wangyujun}@xiaomi.com

Abstract

Speaker adaptation aims to estimate a speaker specific acoustic model from a speaker independent one to minimize the mismatch between the training and testing conditions arisen from speaker variabilities. A variety of neural network adaptation methods have been proposed since deep learning models have become the main stream. But there still lacks an experimental comparison between different methods, especially when DNN-based acoustic models have been advanced greatly. In this paper, we aim to close this gap by providing an empirical evaluation of three typical speaker adaptation methods: LIN, LHUC and KLD. Adaptation experiments, with different size of adaptation data, are conducted on a strong TDNN-LSTM acoustic model. More challengingly, here, the source and target we are concerned with are standard Mandarin speaker model and accented Mandarin speaker model. We compare the performances of different methods and their combinations. Speaker adaptation performance is also examined by speaker's accent degree. **Index Terms:** Speaker adaptation, deep neural networks, LIN, KLD, LHUC

1. Introduction

Speech recognition accuracy has been significantly improved since the use of deep learning models (DLMs), or more specifically, deep neural networks (DNNs) [1, 2]. Various models, such as convolutional neural networks (CNNs) [3, 4], time-delay neural networks (TDNNs) [5], long short-term memory (LSTM) recurrent neural networks (RNNs) [6, 7] and their variants [8, 9] and combinations [10], have been developed to further improve the performance. However, the accuracy of an automatic speech recognition (ASR) system in real applications still lags behind that in controlled testing conditions. This raises the old and unsolved problem called *training-testing mismatch*, i.e., the training set cannot match the new acoustic conditions or fails to generalize to new speakers. Thus a variety of acoustic model representations and adaptation methods have been proposed, to better deal with unseen speakers and mismatched acoustic conditions.

This study specifically focuses on *speaker adaptation*, i.e., modifying a general model, commonly a speaker-independent acoustic model (SI AM), to work better for a specific new speaker, through the same adaptation technique can be applied to other mismatched conditions. The history of acoustic model speaker adaptation can be traced back to the GMM-HMM era [11, 12, 13, 14, 15, 16, 17, 18], while the focus has been shifted to neural networks since the rise of DLMs. Various approaches have been developed for neural network acoustic model adaptation [19, 20, 21, 22, 23, 24, 25, 26, 27, 28] and they can be roughly categorized into three classes: speaker-adapted layer insertion, subspace method and direct model adapting.

*Corresponding author



In the category of speaker-adapted layer insertion, linear transformation, which augments the original network with certain speaker-specific linear layer(s), is a simple-but-effective approach. Common methods include linear input network (LIN) [19, 20], linear hidden network (LIHN) [21], and linear output network (LOH) [20], just to name a few. Among them, LIN is the most popular one. Learning hidden unit contribution (LHUC) [22] is another type of speaker-adapted layer insertion method that makes the SI network parameters to be speaker-specific by inserting special layers to control the amplitude of the hidden layers.

Another category, subspace method, aims to find a low-dimensional speaker subspace that is used for adaptation. The most straightforward application is to use subspace-based features, e.g., i-vectors [23, 24] as a supplement of acoustic features in the neural network for neural model training, or speaker adaptive training (SAT). Another approach, serving the same purpose with auxiliary features, is called speaker roles [25]. A specific set of network units for each speaker is connected and optimized with the original SI network. Note that i-vector-based SAT has become a standard in the training of deep neural network acoustic models [5, 24, 27, 29, 30, 31] as this simple trick can bring small-but-consistent improvement.

A straightforward idea is to use new speaker's data to adapt the DNN parameters directly. Retraining/fine-tuning the SI model using the new data is the simplest way, which is also called retrained speaker independent (RSI) adaptation [19]. To avoid over-fitting, conservative training, such as Kullback-Leibler divergence (KLD) regularization [26] is further introduced. This approach tries to force the posterior distribution of the adapted model to be closer to that estimated from the SI model, by adding a KLD regularization term to the original cross entropy cost function to update the network parameters. Although quite effective, this approach results in an individual neural network for each speaker.

To the best of our knowledge, there still lacks a thorough experimental comparison between different speaker adaptation methods in the literature, especially when the DNN-based acoustic models (AMs) have been advanced greatly since the introduction of these adaptation techniques. In this paper, we aim to close this gap by providing an empirical evaluation of three typical speaker adaptation methods: LIN, LHUC and KLD. Adaptation experiments are conducted on a strong TDNN-LSTM acoustic model (well trained i-vector-based SAT-DNN acoustic model with cMLLR [13, 15]) tested with different size of adaptation data. More challengingly, here, the source and target we are concerned with are standard Mandarin speaker model and accented Mandarin speaker model. We compare the performance of different methods and their combinations. The speaker adaptation performance is also examined by speaker's accent degree. In a word, we would like to provide readers a big picture on the selection of speaker adaptation techniques.

The rest of this paper is organized as follows. In Section 2, we briefly introduce LIN, KLD, LHUC and give a discussion

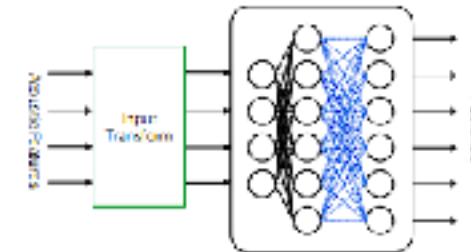


Figure 1: Linear input network.

on their abilities. Next, we describe a series of experiments and report the results in Section 3. Finally, some conclusions are drawn in Section 4.

2. Speaker adaptation algorithms

2.1. LIN

Linear input network (LIN) [19, 20] is a classical input transformation approach for neural network adaptation. As shown in Figure 1, LIN assumes that the mismatch between training and testing can be captured in the feature space by employing a trainable linear input layer which maps speaker dependent speech to speaker independent network (i.e. generic model). The inserted layer usually has the same dimension as the original input layer and is initialized to an identity weight matrix and 0 bias. Unlike other layers of the neural network, linear activation function $f(x) = x$ is used for this additional layer.

During adaptation, standard error back-propagation (BP) is used to update the LIN's parameters while keeping all other network parameters fixed, by minimizing the loss function (e.g., cross entropy, mean square error) of the original AM. After adaptation, each speaker-specific LIN captures the relations between the speaker and the training space. Finally, for each testing speaker, the corresponding LIN is selected to do feature transformation and the transformed vector is directly fed to the original unadapted AM for speech recognition.

2.2. KLD Regularization

As a popular conservative training adaptation technique, Kullback-Leibler divergence (KLD) [26] regularization tries to force the posterior distribution of the adapted model to be closer to that estimated from the SI model. By contrast, the F_2 regularization aims to keep the parameters of adapted model to be closer to those of the SI model.

For acoustic model training, it is typical to minimize the cross entropy (CE)

$$\mathcal{J}_{CE} = -\frac{1}{N} \sum_{n=1}^N \sum_{s=1}^S \hat{p}(y|x_n) \log p(y|x_n), \quad (1)$$

where N is the number of training samples, S is the total number of states, $\hat{p}(y|x_n)$ is the target probability and $p(y|x_n)$ is neural network's output posterior. We usually use a hard alignment from an existing ASR system as the training labels and set $\hat{p}(y|x_n) = \delta(y = s_n)$, where δ is the Kronecker delta function and s_n is the label of n -th sample. By adding the KLD term to Eq. (1) we get the following optimization criterion:

$$\begin{aligned} \mathcal{J}_{CE} &= -(1-\rho)\mathcal{J}_{CE} + \rho \frac{1}{N} \sum_{n=1}^N \sum_{s=1}^S \rho^{\beta} \hat{p}(y|x_n) \log p(y|x_n) \\ &= -\frac{1}{N} \sum_{n=1}^N \sum_{s=1}^S \left[(1-\rho)\hat{p}(y|x_n) + \rho \rho^{\beta} \hat{p}(y|x_n) \right] \log p(y|x_n) \\ &= -\frac{1}{N} \sum_{n=1}^N \sum_{s=1}^S \hat{p}(y|x_n) \log p(y|x_n), \end{aligned} \quad (2)$$

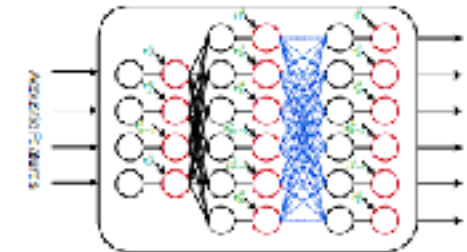


Figure 2: Learning hidden unit contribution.

where ρ is regularization weight and we have defined

$$\hat{p}(y|x_n) \triangleq (1-\rho)\hat{p}(y|x_n) + \rho \rho^{\beta} \hat{p}(y|x_n), \quad (3)$$

By comparing Eq. (1) and Eq. (2), we can find that applying KLD is equivalent to changing the target distribution in the conventional BP algorithm. When $\rho = 0$, we can regard this configuration as RSI, i.e., retaining the SI model directly using the traditional CE loss.

2.3. LHUC

As shown in Figure 2, learning hidden unit contribution (LHUC) [22] modifies the SI model by defining a set of speaker dependent parameters θ for a specific speaker, where $\theta = \{\theta^1, \dots, \theta^L\}$ and θ^l is the vector of speaker dependent parameters for l -th hidden layer. Then the element-wise function $\sigma(\cdot)$ is adopted to constrain the range of θ^l and the speaker dependent hidden layer output can be defined as the following function:

$$h^l = \sigma(\theta^l) \circ e^l(W^{l+1}h^{l-1}), \quad (4)$$

where \circ is an element-wise multiplication and $e(\cdot)$ is typically defined as a sigmoid with amplitude 2, i.e.,

$$\sigma(\theta^l) \triangleq \frac{2}{1 + \exp(-\theta^l)}, \quad (5)$$

to constrain the range of θ^l 's elements to $(0, 2)$.

LHUC, given adaptation data, actually rescales the contributions (amplitudes) of the hidden units in the model without actually modifying their feature receptors. At the training stage, θ is optimized with the standard BP algorithm while keeping all the other parameters fixed for a specific speaker. During the testing stage, the corresponding θ is chosen to constrain the amplitudes of hidden units in order to get more accurate posterior probability for the speaker.

2.4. Discussion and Combination

We compare the three speaker adaptation approaches in terms of adapted parameter size and modification on the AM.

- **Size of Adapted Parameters:** LHUC has minimal adapted parameters, followed by LIN. For KLD regularization, since each speaker has a fully adapted neural network AM, it results in the largest size of adapted parameters.
- **Modification on AM:** In the KLD regularization based adaptation, we do not need to change the original AM network structure, while only changing the loss function. By contrast, we need to adjust the network structure, e.g., inserting layers in the use of LIN and LHUC. However, we need to take extra burden to find an appropriate regularization weight ρ in the KLD regularization based adaptation, which is searched through the validation set.

The three approaches perform network adaptation from different aspects and thus can be integrated to expect some extra



小爱语音识别服务

识别率不足 10%



使用 5 分钟的演讲数据做模型自适应

自适应后的模型

识别率大于 95%



声音不止语音

Human sounds

- Human voice
- Whistling
- Respiratory sounds
- Human locomotion
- Digestive
- Hands
- Heart sounds, heartbeat
- Otoacoustic emission
- Human group actions

Animal sounds

- Domestic animals, pets
- Livestock, farm animals, working animals
- Wild animals

Natural sounds

- Wind
- Thunderstorm
- Water
- Fire

Music

- Musical instrument
- Music genre
- Musical concepts
- Music role
- Music mood

Sounds of things

- Vehicle
- Engine
- Domestic sounds, home sounds
- Bell
- Alarm
- Mechanisms
- Tools
- Explosion
- Wood
- Glass
- Liquid
- Miscellaneous sources
- Specific impact sounds

Source-ambiguous sounds

- Generic impact sounds
- Surface contact
- Deformable shell
- Onomatopoeia
- Silence
- Other sourceless

Channel, environment and background

- Acoustic environment
- Noise
- Sound reproduction

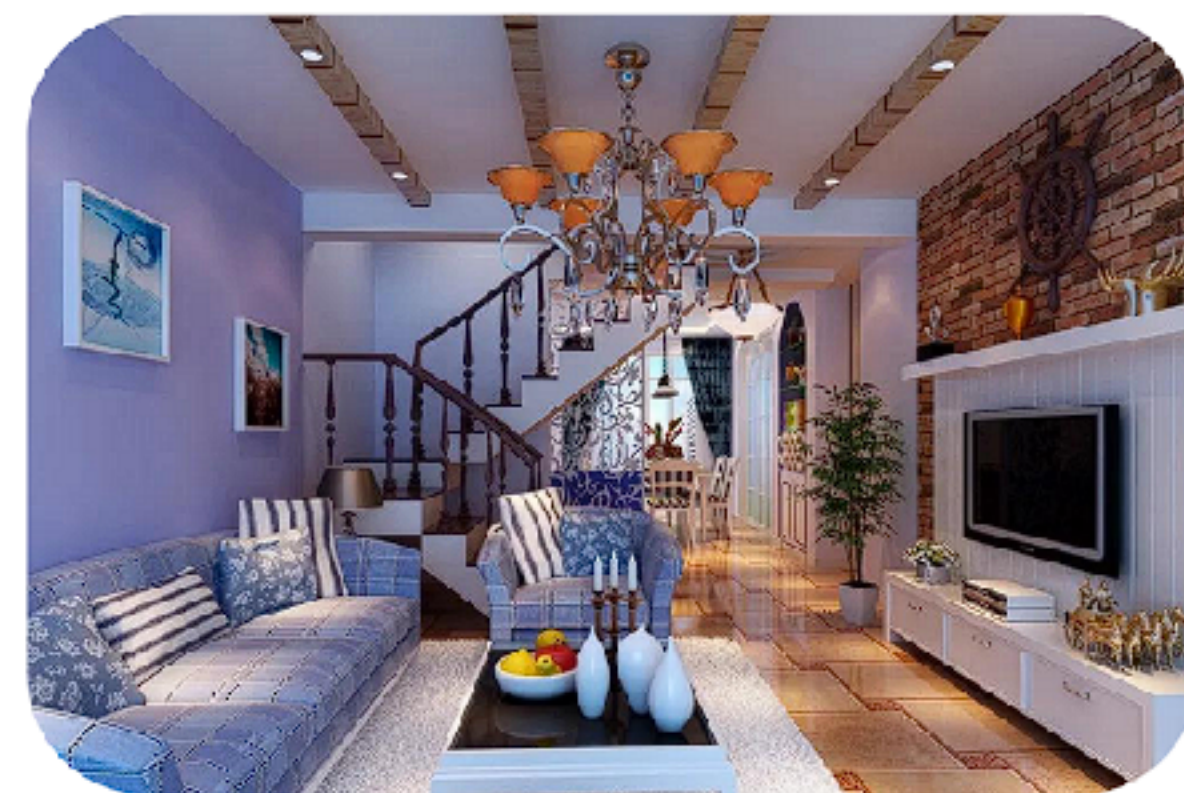
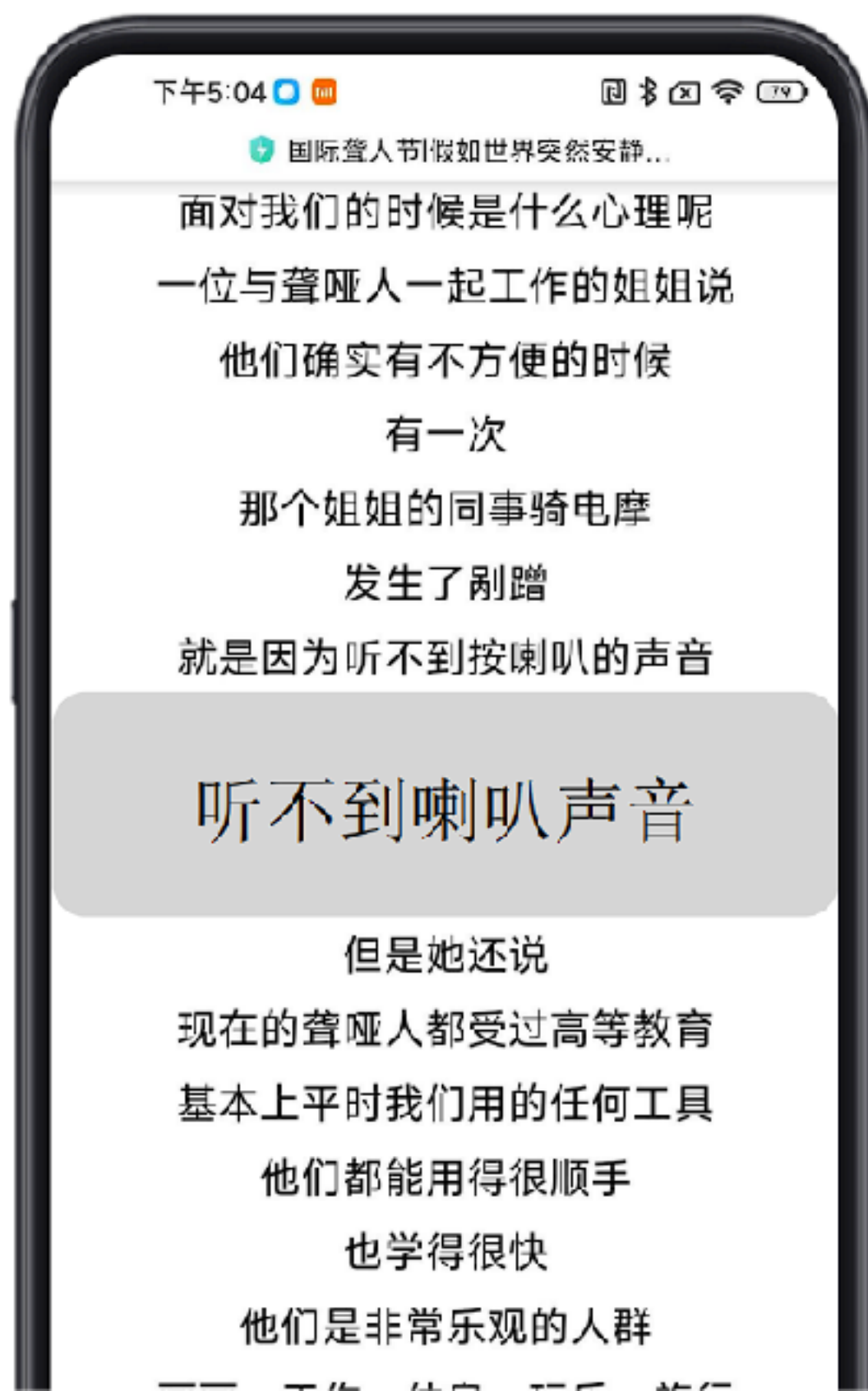




2780万
听力障碍人群

1.18亿
独居老人

在某种程度上
每个人都是“听障人士”





让环境中的每种声音，可以被智能设备感知、识别，并反馈给用户

让听障人士“看到”周围的声音情况

让独居老人“看到”家中的声音

让婴儿的啼哭被爸爸妈妈及时“看到”

让宠物的叫声被主人“看到”

让敲门的声音被里屋的主人“看到”

.....

让声音，被“看到”





小米闻声

声音类别

报警

- 家用报警器 (烟雾、燃气)
- 警笛
- 火警

住宅

- 婴儿啼哭
- 敲门
- 门铃
- 流水

趣味

- 猫叫
- 狗叫

10:52 91%

环境音

您的音箱可识别特定类别的声音并通过小米音箱App推送通知您，以此满足家庭异常情况监控、老年人安全守护、特殊人群无障碍辅助的需求。



XIAOMI Sound
小米高保真智能音箱

- 家用报警器
- 婴儿啼哭
- 火警
- 流水
- 猫叫
- 狗叫

请注意，在可能导致您受伤或位于高风险环境时，不应完全依赖此功能。

注：“家用报警器”声音包含烟雾报警器、燃气报警器及其他家电运行中可能发出的弹响声，音箱检测到此类声音，会按“家用报警器”类别进行通知推送。


睡眠

今天

开启“后台运行无限制”以更准确的记录睡眠状况

鼾声梦话

打鼾 4分钟 梦话 0段



00:22 03:00 05:00 07:11

00:00 00:00

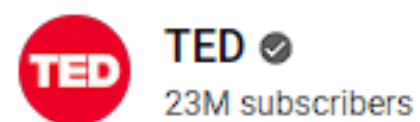
- 04:09 鼾声片段 27秒
- 04:10 鼾声片段 42秒
- 04:16 鼾声片段 38秒



健康



The power of introverts | Susan Cain



Subscribe

393K

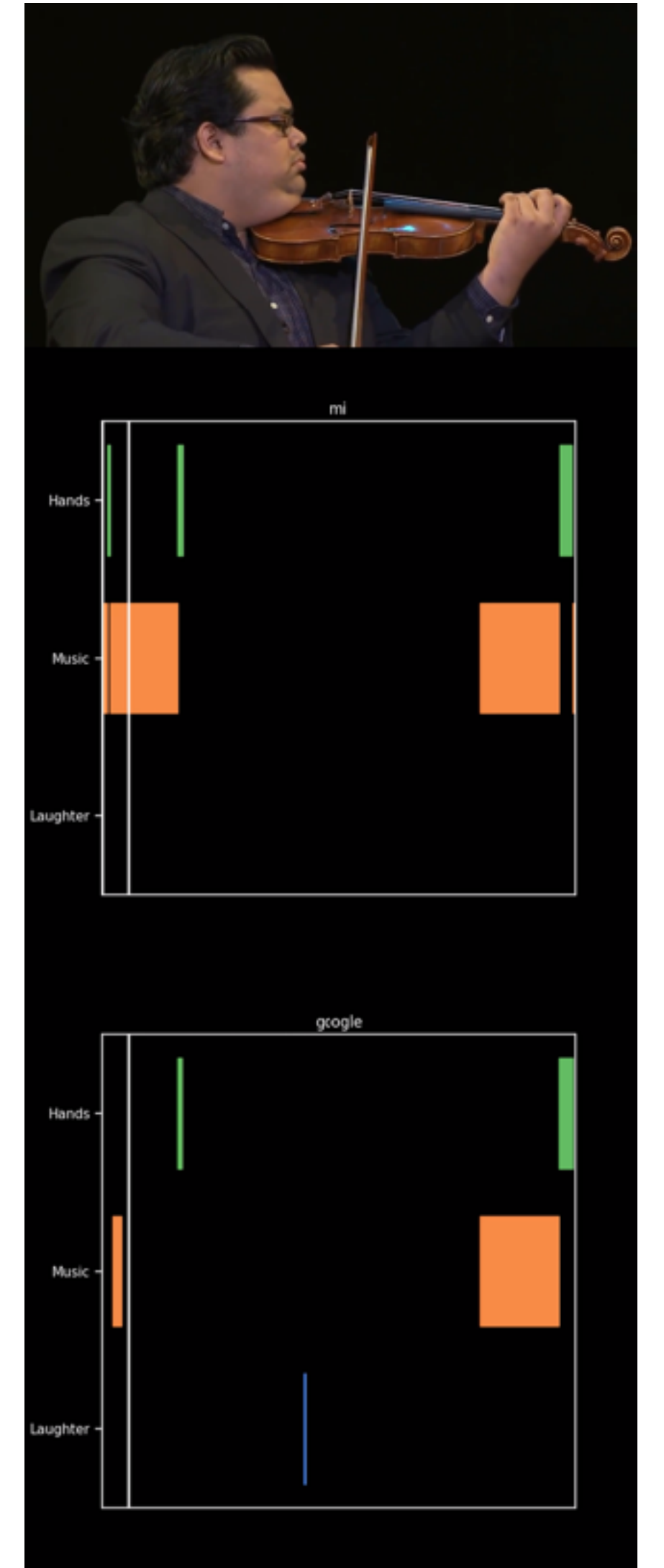


Share

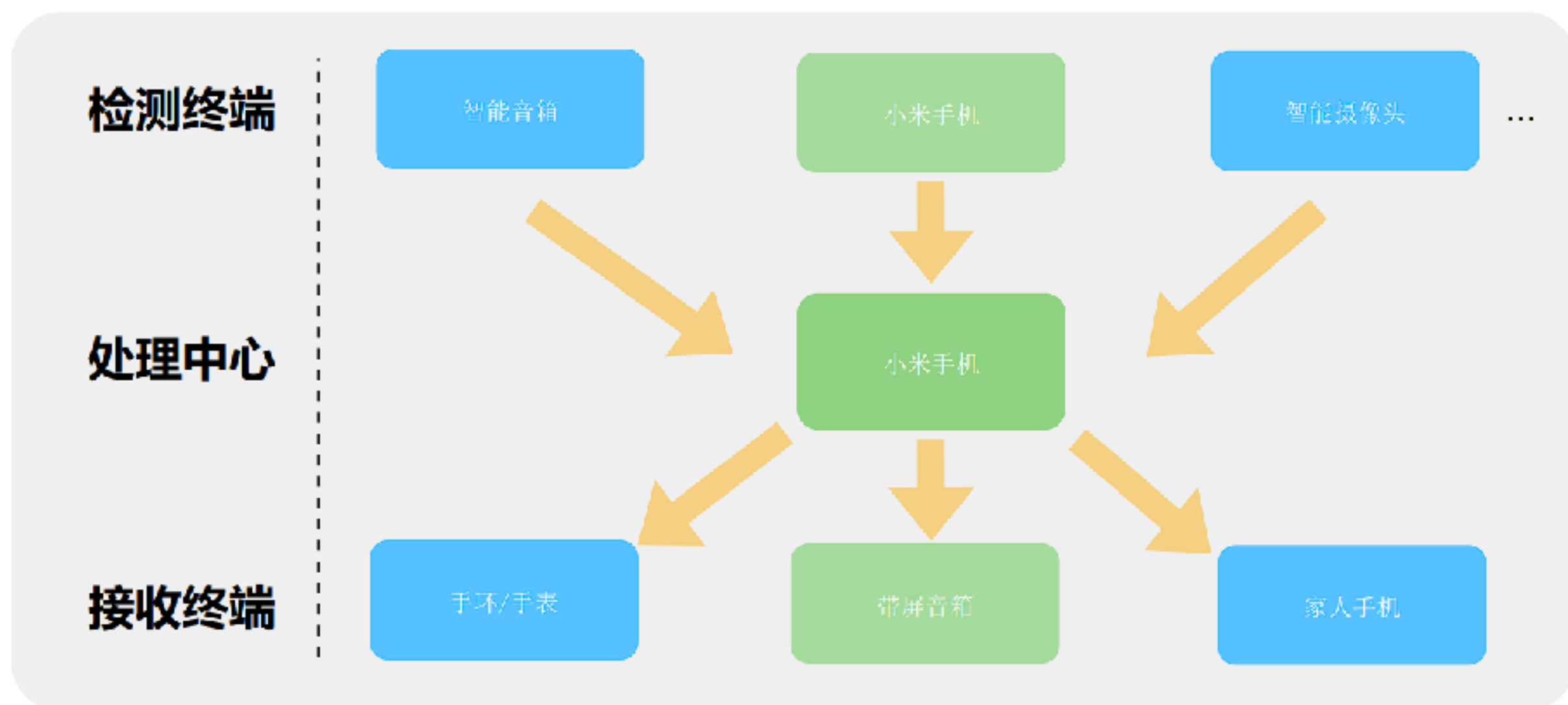
Download

Clip

Save



多端协同



分层通知，减少打扰

当同一房间内多设备检测到相同声音时，则只下发一条通知
当不同房间多设备检测到多次相同声音时，则多次下发多条通知

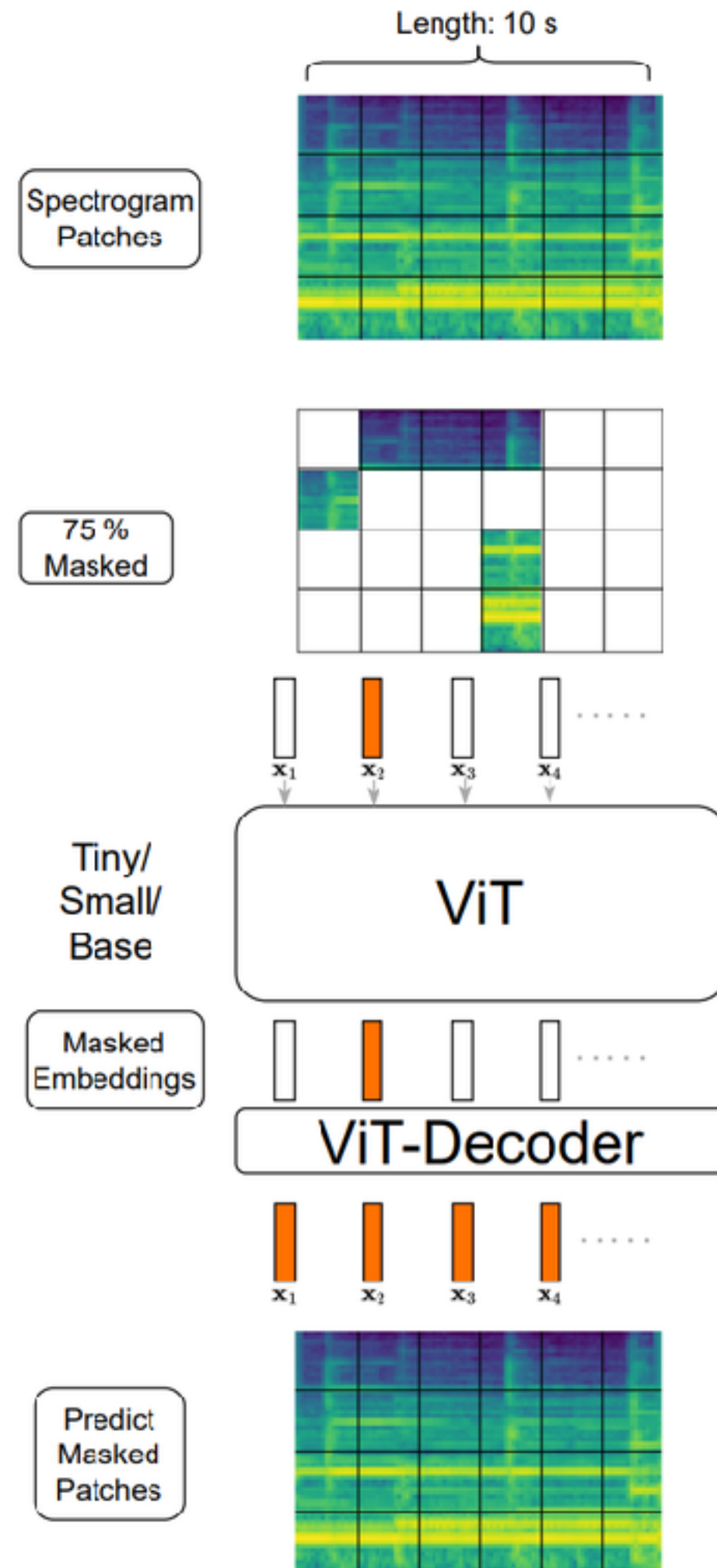
重点声音，高优通知

婴儿啼哭、烟雾警报等重要声音，迅速、多次、重点通知
宠物叫声、水流声，可在发生多次后再通知

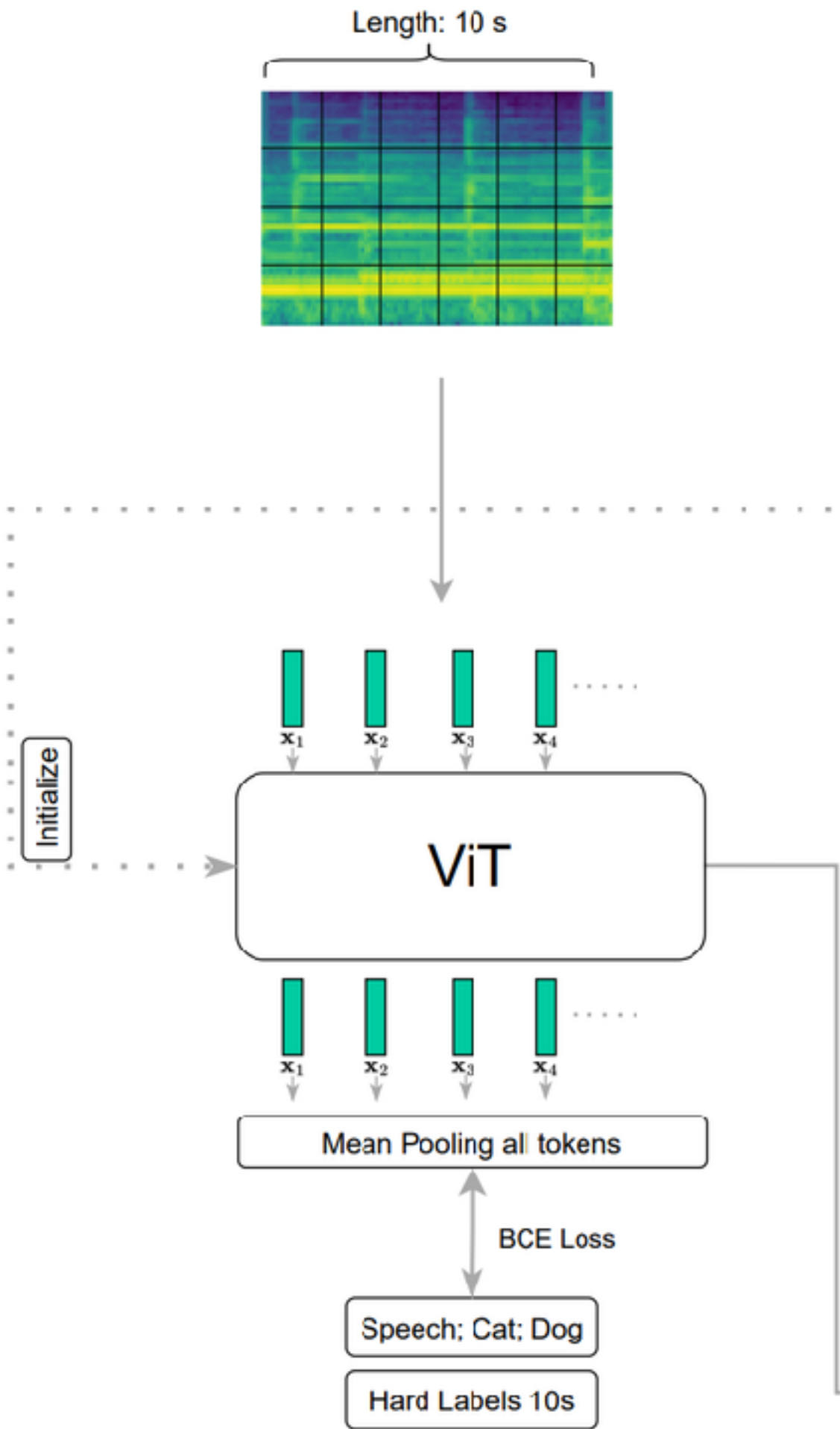


模型训练流程

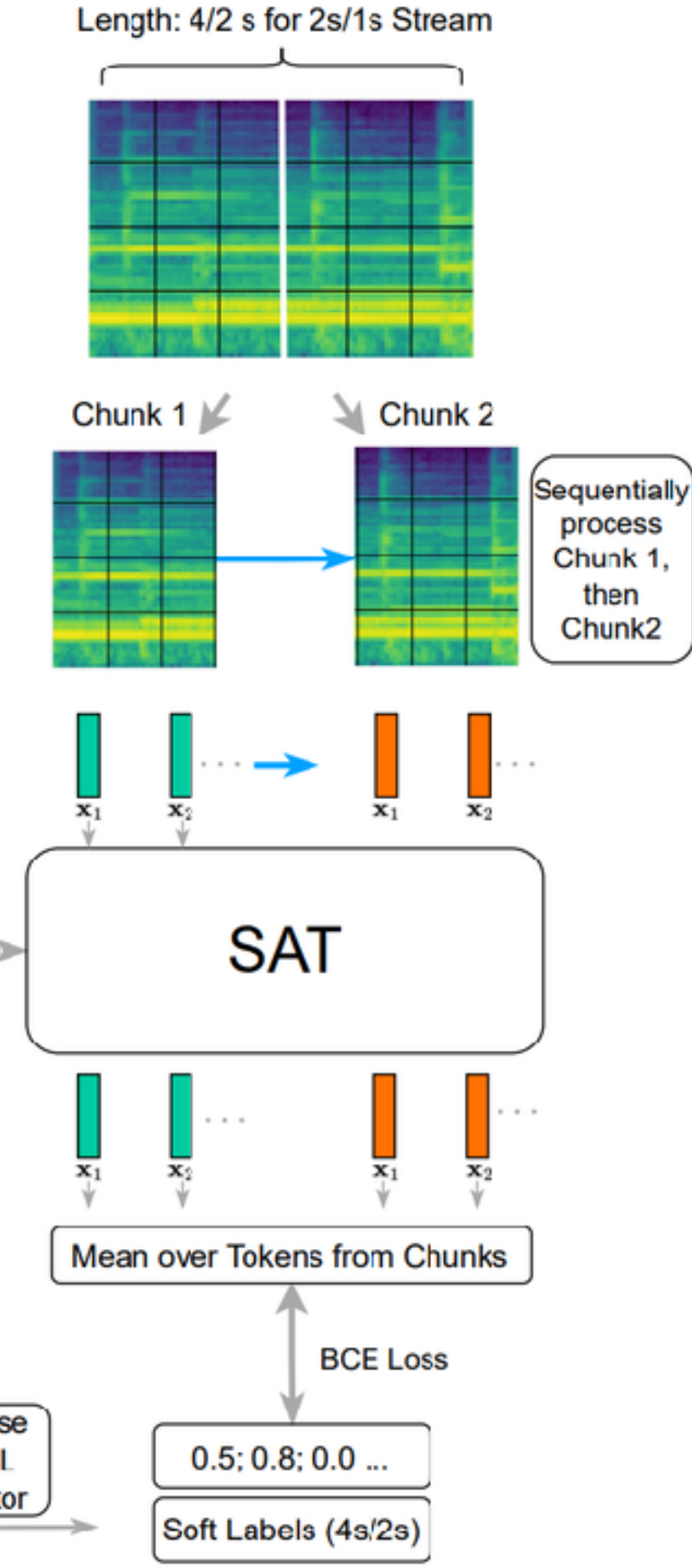
1. Audio MAE pretraining



2. Full-context Training



3. SAT Training



arXiv:2305.17834v1 [cs.LG] 29 May 2023

Streaming Audio Transformers for Online Audio Tagging
 Heinrich Dinkel, Zhiyong Yan, Xinyang Wang, Anbo Zhang, Yijun Wang
 Xiaomi Corporation, Beijing, China
 {hinkelheirich, yanziyong, wangxinyang1, zhangjunbo1, wangyijun}@xiaomi.com

Abstract
 Transformers have emerged as a prominent model framework for audio tagging (AT), boasting state-of-the-art (SOTA) performance on the widely-used Audioset dataset. However, their impressive performance often comes at the cost of high memory usage, slow inference speed, and considerable model delay, rendering them impractical for real-world AT applications. In this study, we introduce streaming audio transformers (SAT) that combine the vision transformer (ViT) architecture with Transformer-XL like chunk processing, enabling efficient processing of long-range audio signals. Our proposed SAT is benchmarked against other transformer-based SOTA methods, achieving significant improvements in terms of mean average precision (mAP) at a delay of 2s and 1s, while also exhibiting significantly lower memory usage and computational overhead. Checkpoints are publicly available <https://github.com/XiaoHeinrich/SAT>.

Index Terms: Audio Tagging, Vision Transformer, Streaming Inference, In-line Inference

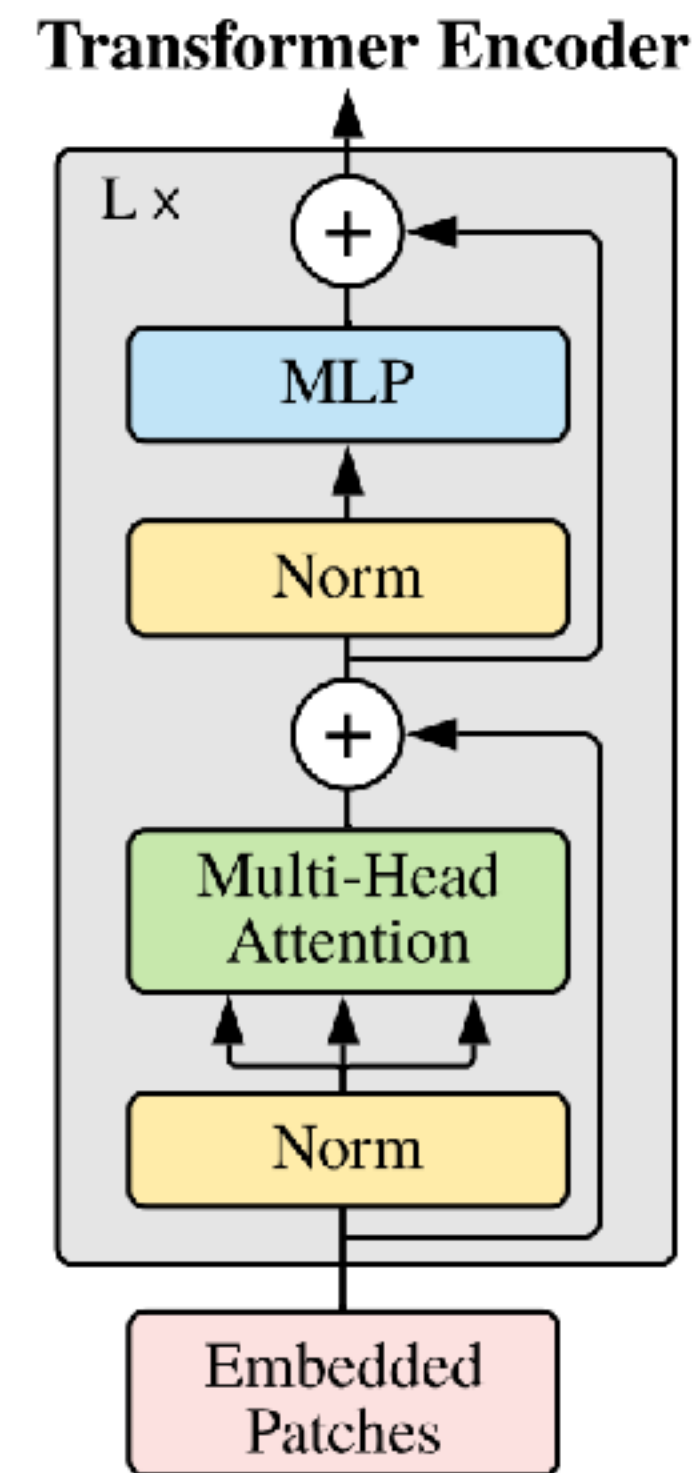
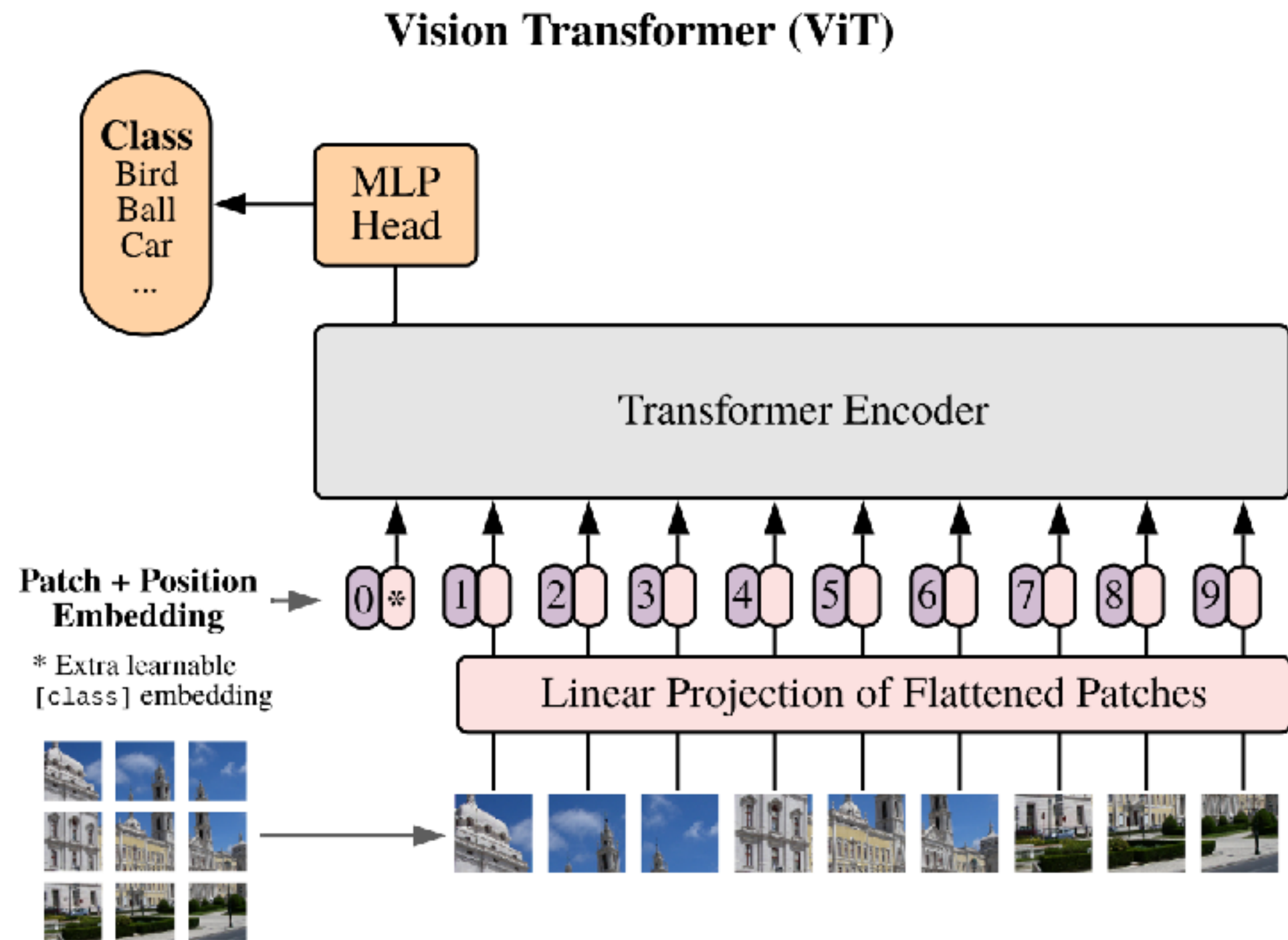
1. Introduction
 Audio tagging (AT) is a task that aims to label specific audio content into a fixed set of sound event classes, e.g., dog barking or people speaking. Applications of AT systems include aid for the hearing impaired, smart cities and homes [1] and general monitoring of sounds [2]. More recently, AT systems have found applications on smartphones and smart speakers as a hearing aid for the user. The transformer model, originally introduced in [3], which uses self-attention as its core building block, has become a popular method to achieve excellent performance for AT, however, the deployment of transformer architectures in real-world scenarios has been largely neglected. Previous works using Vision Transformer (ViT) based models such as in [4, 5, 6, 7, 8, 9, 10, 11] are optimized towards offline usage with global context of 10s. Unfortunately, this approach results in a model response time (delay) of at least 10s. In our work, we reduce delay as the amount of data that a model needs to process before generating an output. Many transformer architectures in AT suffer from a high memory requirement due to their quadratic self-attention complexity, which depends on the amount of data processed at once [10]. However, real-world applications are online, meaning that a model needs to return results as quickly as possible with a minimal delay while having access to a limited context (i.e., 1s). Although one may easily enable “online” inference by reprocessing a 10s audio segment every e.g., 1s, this practice is inefficient, particularly when leveraging large transformer-based models [4]. To address this challenge, streaming inference algorithms aim to compute outputs efficiently without the need for recomputation by leveraging caching of previous results. This work widely focuses on optimizing transformer-based models towards streaming inference, since traditionally used 2-dimensional convolutional neural networks (CNNs) are hard to make streaming [12]. We point out three essential prerequisites of AT models for real-world deployment, namely: (I) A minimal delay in terms of data necessary to output a label, typically on the order of 1-2 seconds. (II) A small memory footprint and low computational complexity. (III) Robust and reliable performance. While there exist many works in literature that tackle the problem of low delay [13], reducing memory footprint [7, 14] and improving performance [15, 4, 6], an comprehensive investigation has yet tackled all these issues. Thus, this work proposes streaming audio transformers (SAT), aimed at real-world usage of transformers for AT. Our motivation for this work is twofold. Firstly, it would improve compatibility between AT models and other audio subfields that are streaming, such as automatic speech recognition [16, 17], keyword spotting [18, 19] and voice separation [20, 21]. Secondly, when deployed on stationary hardware like smart speakers, SAT models could act as an anomaly detector for long recurring sounds, differentiating between harmless and potentially harmful events, such as a single beep from a fire alarm versus continuous beeping. As we empirically demonstrate in this work, standard AT models struggle to continuously predict sound events (Section 5.4). Our contributions are: (I) We experiment with three standard ViT models (Tiny, Small, Base), and optimize the training pipeline for AT, aiming to reduce their memory consumption and decrease their floating-point operations per second (Flops). (II) Based on these three models, we introduce streaming (SAT) variants, denoted as SAT-T (Tiny), SAT-S (Small) and SAT-B (Base). We compare these models with other transformers in the literature and find significant performance improvements to real-time inference.

2. Vision Transformers for Audio Tagging
 Transformers were first proposed for machine translation in [3] and quickly became the state-of-the-art (SOTA) approach within the field of natural language processing (NLP) and later [22] the Vision Transformer (ViT) has been proposed as an algorithm to the computer vision domain. Then, ViT based transformers were used in AT where images were replaced with two-dimensional spectrograms [4, 23]. The core idea of the ViT framework is the patchification operation, where an input spectrogram $S \in \mathbb{R}^{T \times F \times C}$ is first split into N non-overlapping patches of dimension d via a 2-dimensional correlation with a kernel-size P and stride P as:

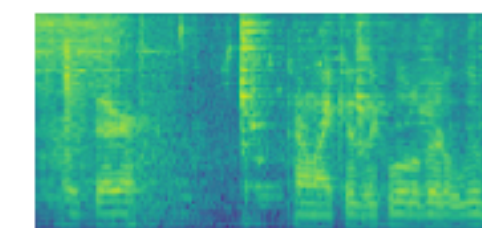
$$X = \text{Conv2DS}(P, P) = (x_1, x_2, \dots, x_N). \quad (1)$$



模型结构 - Vision Transformer (ViT)



图片

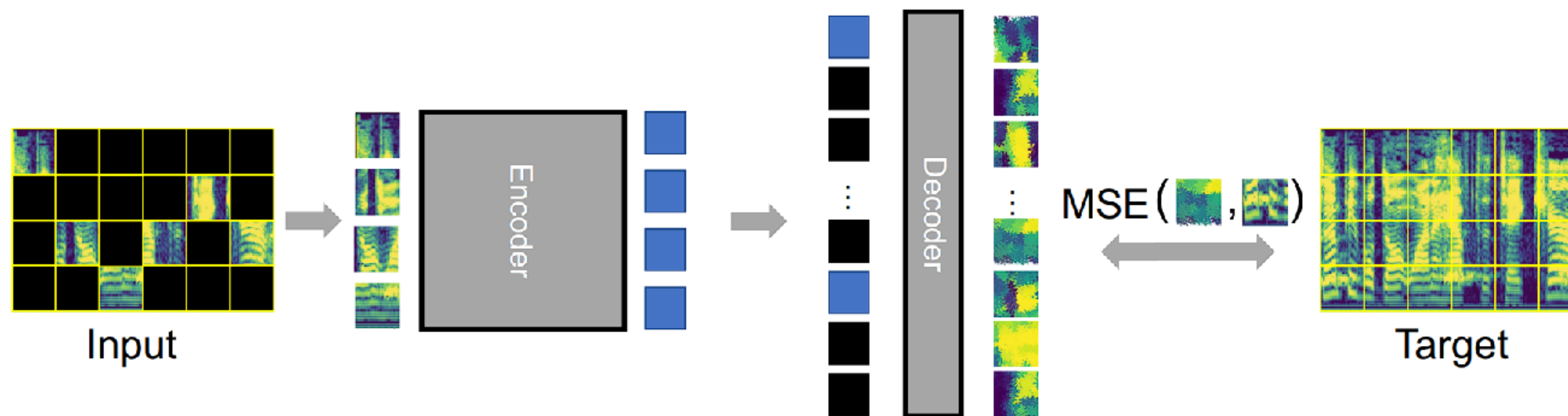


声音

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

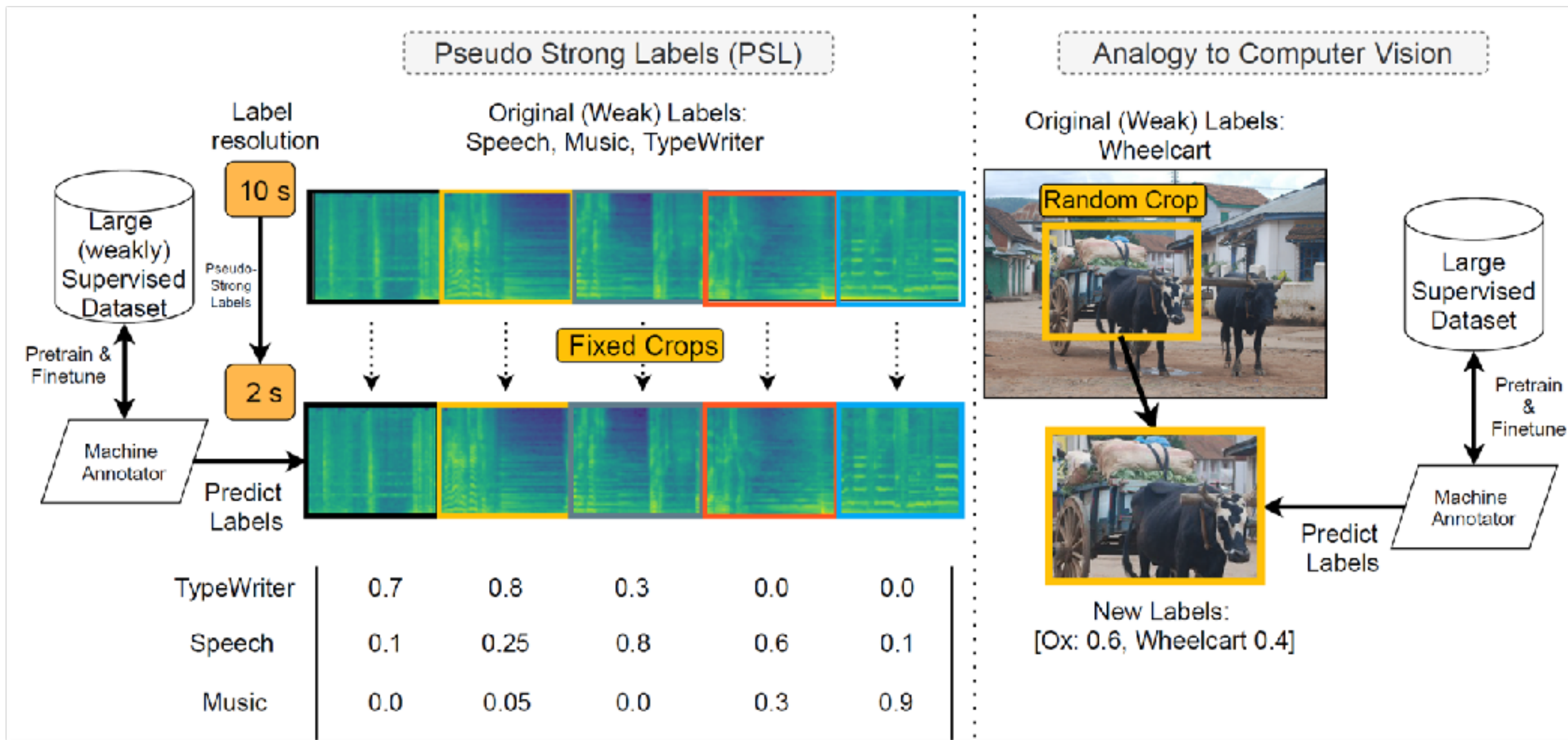


自监督训练 – Audio MAE



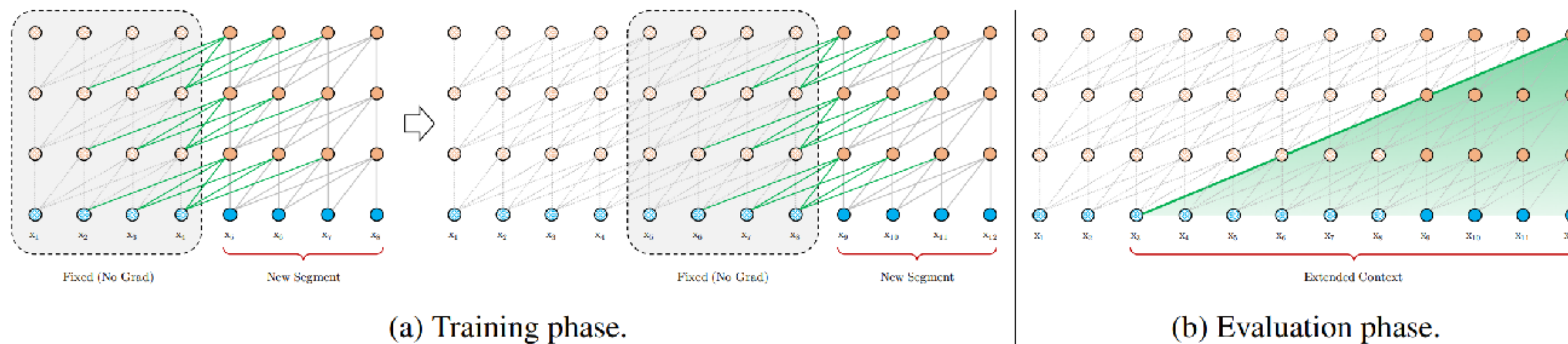
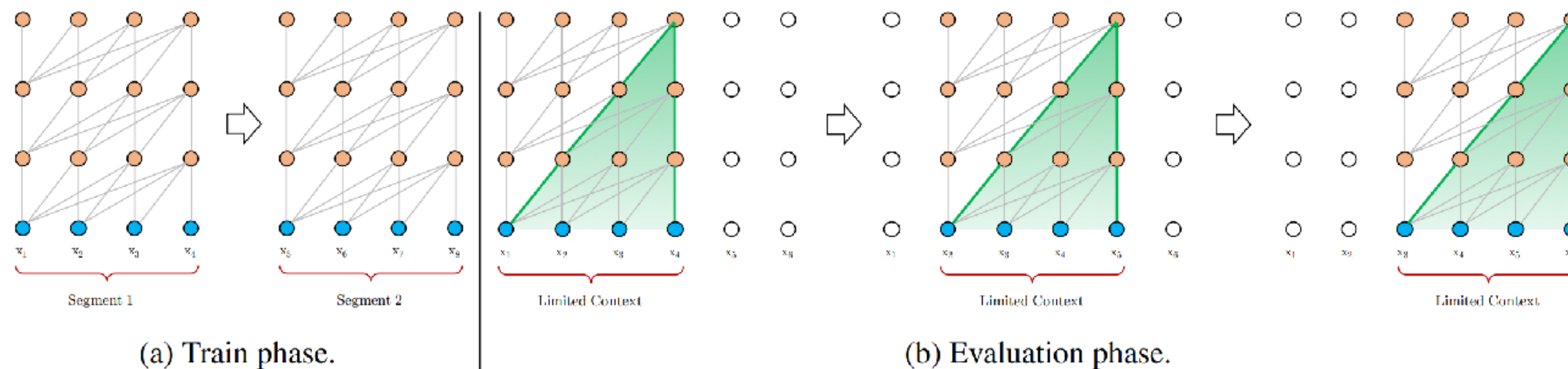
已收集时长约 31 年的训练数据

伪强标签生成 - PSL



	AST-10s	AST-2s	(Ours)
mAP	45.9	39.7	44.2
参数量	86 M	86 M	< 6 M
峰值内存	2.2 G	2.2 G	52 M
延迟	10 s	2 s	2 s
推理速度 (小米10至尊版)	> 200 ms	> 200 ms	~ 2 ms

流式训练





自定义声音识别



使用 iPhone 识别声音

iPhone 可以持续听取某些声音 (如婴儿哭声、门铃或汽笛声) 并在识别出这些声音时通知你。

【注】当你可能受到伤害或受伤、在高风险或紧急情况下或者导航时, 不要依赖 iPhone 识别声音。

设置声音识别

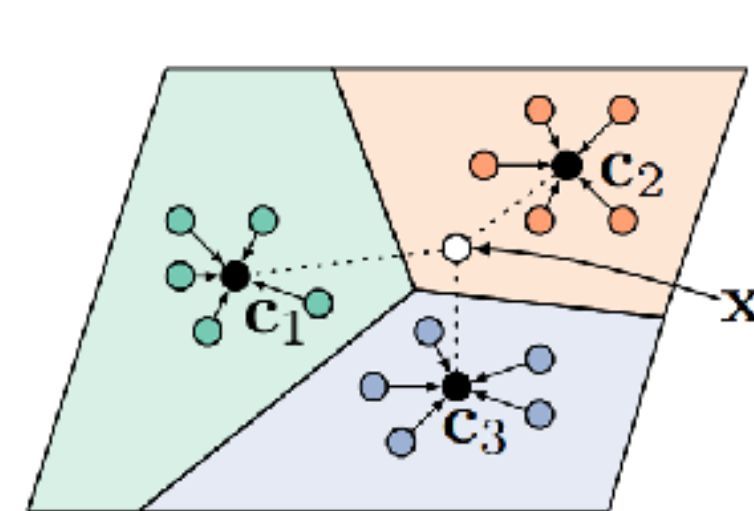
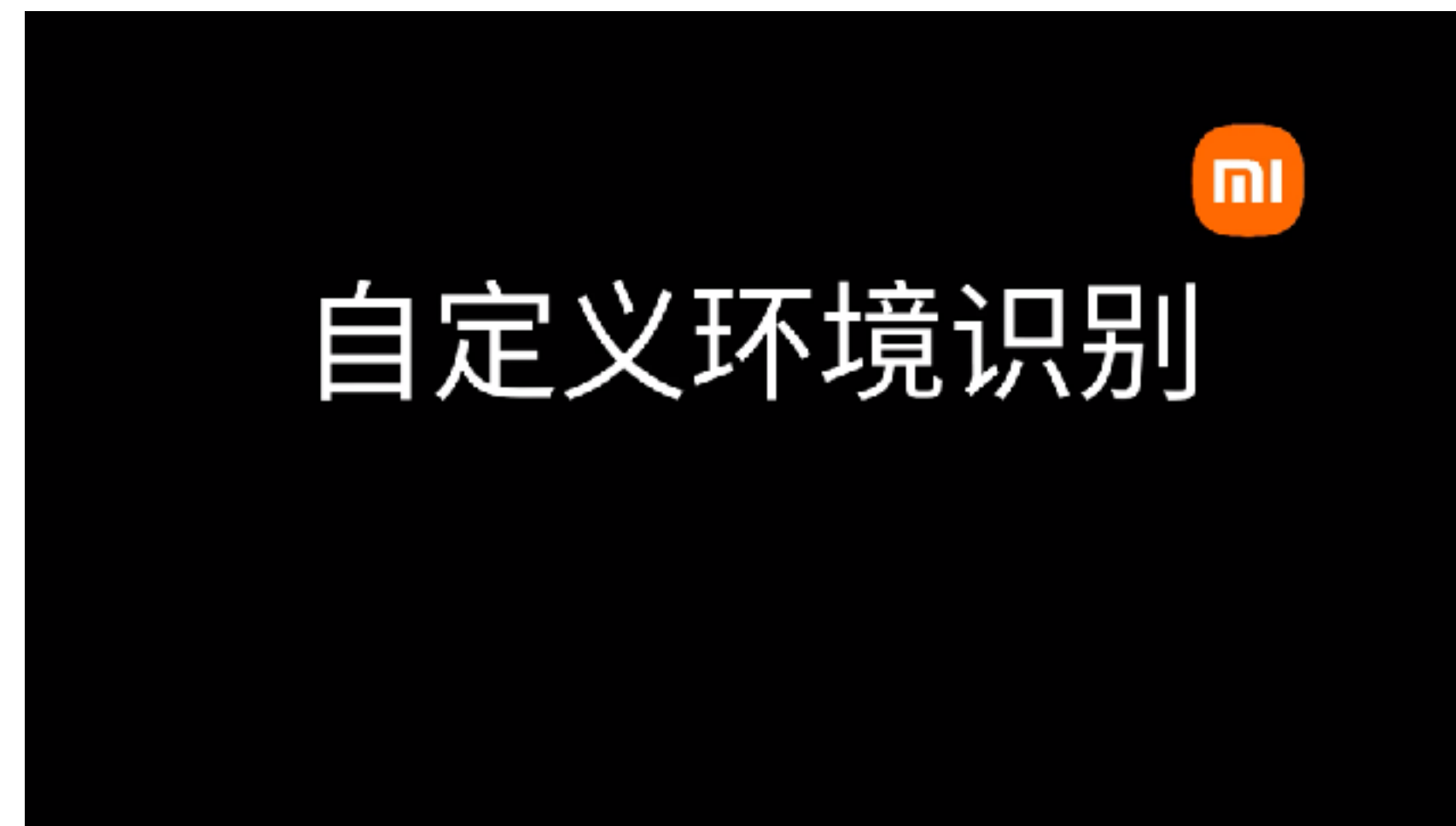
1. 前往“设置”>“辅助功能”>“声音识别”, 然后打开“声音识别”。
2. 轻点“声音”, 然后打开想要 iPhone 识别的声音。

💡【提示】若要快速打开或关闭“声音识别”, 请使用控制中心。

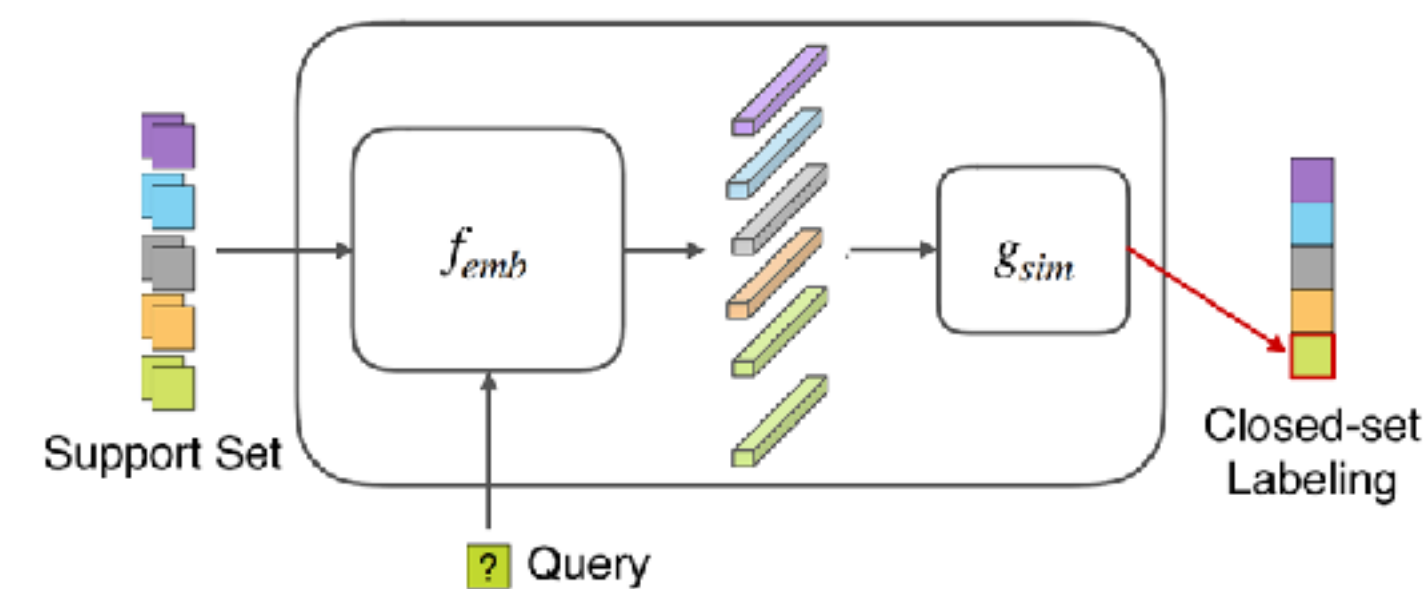
添加自定义警报器、家电声或门铃声

如果声音未自动识别, 你还可以设置 iPhone 以识别自定义的警报器、家电声或门铃声。

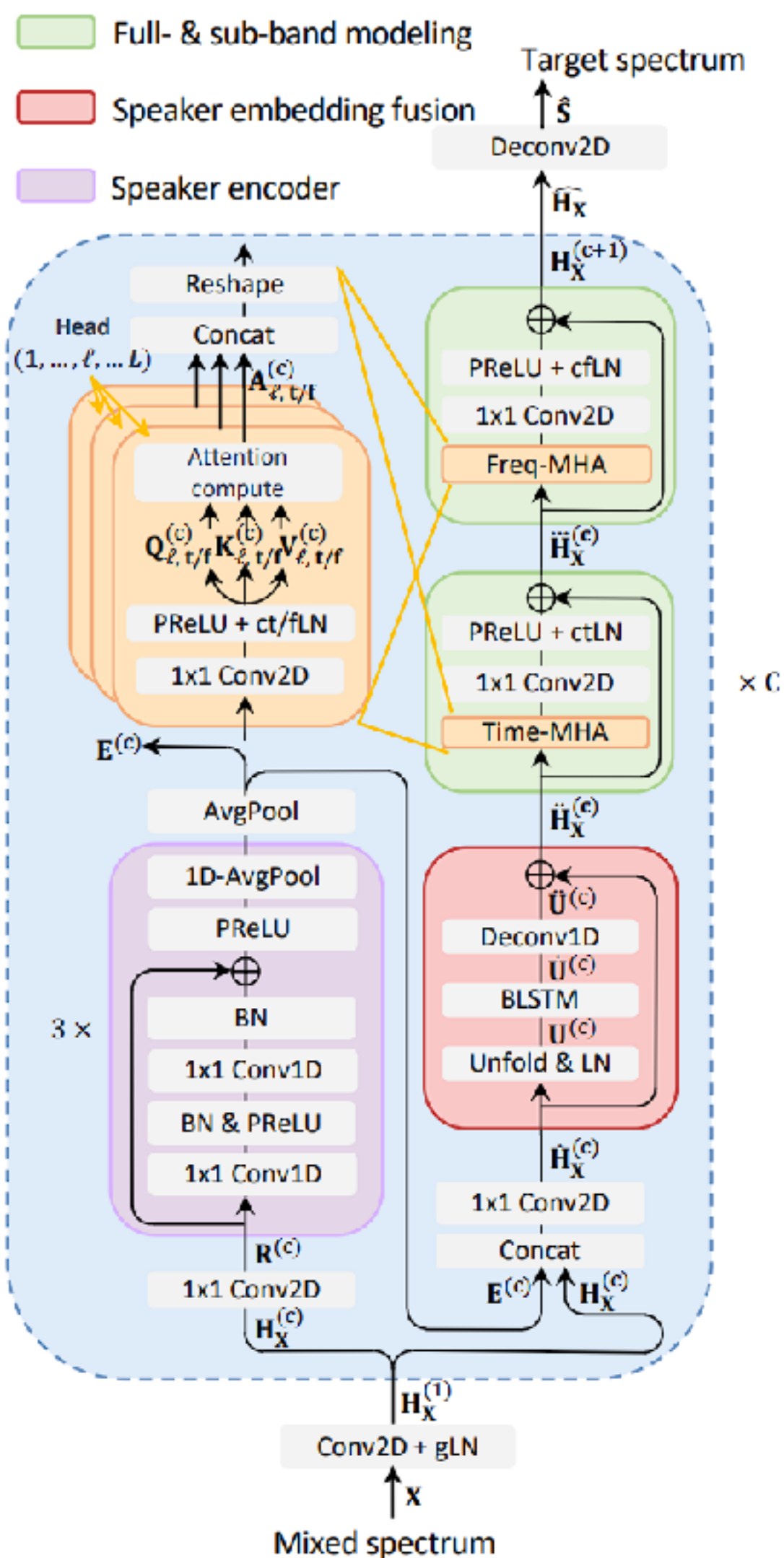
1. 前往“设置”>“辅助功能”>“声音识别”>“声音”。
2. 轻点“自定义警报器”或“自定义家电声或门铃声”, 然后输入一个名称。
3. 警报器、家电或门铃准备就绪后, 将 iPhone 靠近声源并尽量减少背景噪声。
4. 轻点“开始听取”, 然后按照屏幕指示操作。



Few-shot Learning



近距离干扰下的目标语音提取



Focus the Sound around You: Monaural Target Speaker Extraction via Distance and Speaker Information

Jiubin Luo^{1*}, Peng Wang^{2*}, Heinrich Dinkel², Jun Chen¹, Zhiyong Wu¹, Kangqing Wang², Zhiyong Yan², Jianbo Zhang², Yujun Wang²

¹Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
²Xiaomi Inc., Beijing, China

ljub@sgis.tsinghua.edu.cn, pengw@xiaomi.com

Abstract

Previously, Target Speaker Extraction (TSE) has yielded out-standing performance in certain application scenarios for speech enhancement and source separation. However, obtaining auxiliary speaker-related information is still challenging in noisy environments with significant reverberation. Inspired by the recently proposed distance-based sound separation, we propose the near-sound (NS) extractor, which leverages distance information for TSE to reliably extract speaker information without requiring previous speaker enrollment, called speaker embedding self-enrollment (SSE). Full- & sub-band modeling is introduced to enhance our NS-Extractor's adaptability towards environments with significant reverberation. Experimental results on several cross-dataset scenarios demonstrate the effectiveness of our improvements and the excellent performance of our proposed NS-Extractor in different application scenarios.

Index Terms: target speaker extraction, distance-based sound separation

1. Introduction

Target Speaker Extraction (TSE) [1], also known as Target Speech Extraction, is an essential task in the field of audio processing that involves separating a speech signal of a specific speaker from an audio mixture containing multiple speakers. This task has become increasingly important in recent years with the rise of various speech-based applications such as speech recognition [2], speaker verification [3], and audio conferencing. While band speech separation (BSS) is limited by permutation-variant training (PT) [4], ISL methods face no such restriction. Moreover, while ISL can extract the desired speaker's speech directly, BSS outputs several speech signals from different speakers, which requires manual selection. Nevertheless, TSE has a disadvantage: auxiliary information related to the input speaker or such as enrolled voice [5, 7] or lip movements [8–10] are required in advance. Typically, this necessitates attending additional resources and encroaching upon the privacy of the information involved.

Recently, [11] proposed distance-based sound separation (DSS), which can separate monaural audio sources by the perceived distance (due to reverberation) between a listener and a sound emitter. DSS produces two audio signals: one from within a fixed threshold distance ("near") and another from outside the distance ("far"). Currently, DSS may face certain limitations in practical applications. First, the threshold distance for separation cannot be arbitrarily changed during inference, which might result in having multiple "near" sources due to an

intrusive sound source coming into the threshold distance range. As an example, within a meeting, multiple sources might be of equal distance to the microphone, which the approach in [11] is unable to separate. Furthermore, due to the heavy reliance on the reverberation effect, distance-based separation is limited to smaller rooms with a longer reverberation time (RT60), while many offices are in large rooms with a fast reverberation effect. Lastly, previous works based on LSTM [12] can be further optimized to use more modern separation models, which could significantly enhance the user experience. Our work is inspired by the human perception of the cocktail party problem, where humans can selectively focus on a specific sound source (i.e., speaker) if it is clear to them, while still filtering noise from far away sources. Thus we believe that if we incorporate this distance-based source separation into TSE, we can achieve a more potent separation performance.

Although separating mixed audio signals with and without reverberation may appear to be similar tasks, there are significant differences between the two in practice. Reverberation can cause several issues in speech modeling [13], including: (a) Create echoes that overlap with the original speech signal; (b) Dampen the high frequency components of the speech signal; (c) Introduce a delay between the original speech signal and the reverberant sound. All these may lead to a more difficult understanding of speech. Therefore, when conducting TSE in a reverberant environment, a different approach must be taken compared to regular TSE.

While time-domain approaches have seen success on commonly used benchmark datasets such as MUSDB18 [14], some of them such as Cows-TasNet [15] generally perform poorly when faced with reverberant audio [16]. This performance decay has been analyzed in [17], where time-frequency (spectral) domain frameworks have been seen to offer superior separation performance. Additionally, it was indicated that a sub-band model is capable of modifying the reverberation effect by focusing on the temporal evolution of the narrow-band spectrum in the results of [18].

In this work, we propose the Near-Sound Extractor (NS-Extractor), a TSE model combining full-, sub-band modeling and speaker embedding self-enrollment (SSE). NS-Extractor utilizes the perceived distance to the target speaker as a cue to extract a self-enrolled speaker embedding that represents the voice print of the target speaker, which is then used for further extraction. Full- and sub-band modeling are integrated to maintain greater stability in scenarios of performance. Experimental results show that our proposed NS-Extractor not only outperforms the baseline in terms of signal and perceptual quality but also exhibits superior performance in more complex scenarios.

* Equal contribution.
† Corresponding author.



近期发表论文



- A Lightweight Approach for Semi-supervised Sound Event Detection with Unsupervised Data Augmentation
- A Contrastive Semi-Supervised Learning Framework For Anomaly Sound Detection



- An Empirical Study of Weakly Supervised Audio Tagging Embeddings for General Audio Representations



- Pseudo Strong Labels For Large Scale Weakly Supervised Audio Tagging



- UniKW-AT: Unified Keyword Spotting and Audio Tagging



- Unified Keyword Spotting and Audio Tagging on Mobile Devices with Transformers



- Focus the Sound around You: Monaural Target Speaker Extraction via Distance and Speaker Information



思考和结语

- 技术去帮助障碍人群的同时，无障碍也提供了对技术的一个极致测试场景
- 解决障碍人士日常的需求的同时，也解决了普通人类类似不便场景的需求
- 不便场景往往给预研探索提供了落地空间



谢谢!

