Personalized User-Defined Keyword Spotting in Household Environments: A Text-Audio Multi-Modality Approach

Zhiqi Ai, Zhiyong Chen, Xinnuo Li, Shugong Xu

School of Communication & Information Engineering (SCIE), Shanghai University

{aizhiqi-work, zhiyongchen, bingpohun, shugong}@shu.edu.cn

Abstract

In this paper, we introduce Personalized User-Defined Keyword Spotting (PUKWS), a novel pipeline specifically designed for enhancing household environments by integrating user-defined keyword spotting (KWS) with open-set speaker identification (SID) into a cascading dual sub-system structure. For KWS, we present multi-modal user-defined keyword spotting (M-UDKWS), a novel approach that leverages multi-modal prompts for text-audio multimodal enrollment, and optimizes phonetic and semantic feature extraction to synergize text and audio modalities. This innovation not only stabilizes detection by reducing mismatches between query audio and support text embeddings but also excels in handling potentially confusing keywords. For open-set SID, we adopt advanced open-set learning techniques to propose speaker reciprocal points learning (SRPL), addressing the significant challenge of being aware of unknown speakers without compromising known speaker identification. To boost the overall performance of the PUKWS pipeline, we employ a cutting-edge data augmentation strategy that includes hard negative mining, rule-based procedures, GPT, and zero-shot voice cloning, thereby enhancing both M-UDKWS and SRPL components. Through exhaustive evaluations on various datasets and testing scenarios, we demonstrate the efficacy of our methods. PUKWS showcases a 34.8% improvement over existing baselines, significantly enhancing usability in household settings.

Index Terms: user-defined keyword spotting, open-set speaker identification, few-shot learning, data mining

1. Introduction

Personalized user-defined keyword spotting (PUKWS), which includes customized word detection and specific speaker verification, has become more prevalent in people's daily lives through smart terminals and in-vehicle devices [1]. Duration enrollment, users will repeatedly read their preferred phrases, and the device records both the content and speaker information of these audio clips. In the inference stage, the device compares the input audio with enrollment information to determine whether it needs to be woken up [2, 3].

Currently, the most prevalent approach is to combine keyword spotting (KWS) and speaker verification (SV) as a cascade system, where they are independently executed within the pipeline [3, 4, 5, 6, 7]. In general, low-cost KWS will constantly detect whether the target phrase appears in continuous speech. Once triggered, relevant speech segments are fed into the high-cost SV for identity authentication [1, 2]. Another approach involves text-dependent speaker verification (TD-SV) [8]. Joint optimization enables the integration of phoneme and speaker information, which improves the performance of SV [3, 9, 10].

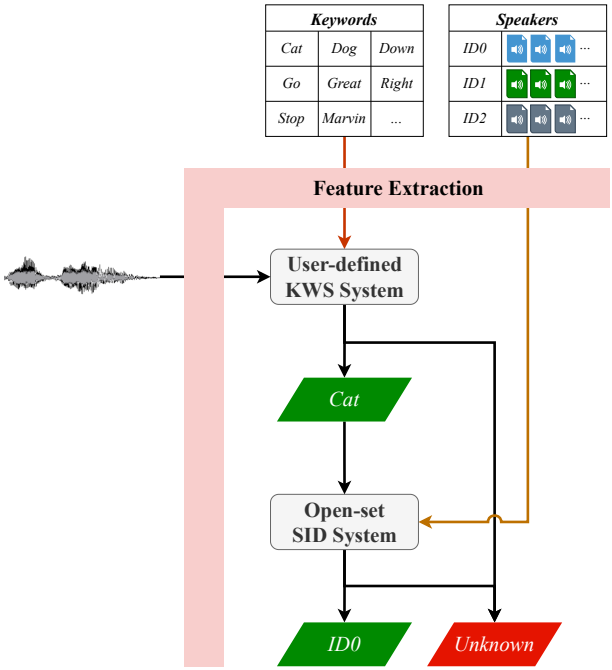


Figure 1: *Overview of the personalized user-defined keyword spotting system.*

However, the current KWS and SV have several limitations. On the one hand, they are text-dependent, limiting the flexibility of customized keywords [2, 3, 6, 9, 10]. On the other hand, SV focus primarily on one-to-one speaker verification rather than the more complex and useful one-to-many speaker identification (SID), lacking optimization for specific scenarios or acoustic domains [11, 12, 13, 14, 15].

Historically, Large Vocabulary Continuous Speech Recognition (LVCSR) has been utilized in user-defined keyword spotting systems [5], but its efficiency is questioned. To address its inefficiency, researchers have focused on the Query-by-Example (QbyE) approach [16, 17]. A mainstream solution is Query-by-Audio (QbyA), which detects keyword spotting by efficient template matching based on acoustic features extracted from speech, such as bottleneck feature and phoneme posterior probability [1, 5], or acoustic word embeddings [18, 19, 20]. In order to improve the robustness of the system to the variation of audio samples, some recent works propose Query-by-Text (QbyT) [21, 22, 23, 24, 25], which extracts text and audio features separately and discriminates the presence of target words in speech based on the monotonicity matching of phonemes and

speech.

A significant challenge in current researches is the lack of ability to effectively utilize multi-modal enrollment data. Considering the usability of speech and text for accurate keyword spotting, we propose a multimodal keyword spotting method called multi-modal user-defined keyword spotting (M-UDKWS). We propose to extract the features of enrolled text and template audio respectively, match them with the query audio using a pattern extractor module, and then decide whether the query audio contains the target word or not by using a pattern discriminator. Additionally, we use the pre-trained G2P [21] and DistilBERT [26] based text content modeling modules to extract the *phonetic* and *semantic* features of the enrolled text, respectively. This approach significantly enhances M-UDKWS's capability to distinguish between homophones and synonyms, thereby improving performance in scenarios with potentially confusing keywords.

In household environments, personalized KWS heavily relies on accurately identifying speakers. These models, designed for *speaker identification* adopt a few-shot learning approach for precise recognition of in-house speakers, effectively utilizing target speaker voice print characteristics in the household [27]. A notable gap is the difficulty in detecting *unknown* speakers while focusing on *known* speakers identification–a challenge known as open-set speaker identification [13]. To bridge this gap, we embrace the advanced open-set learning approach [28], introducing a novel training method, speaker reciprocal points learning (SRPL), tailored for robust open-set SID in household settings.

By integrating M-UDKWS and SRPL, we establish the personalized user-defined keyword spotting system (PUKWS), as illustrated in Figure 1. To further bolster the performance of the PUKWS. We propose a data augmentation strategy with multiple advanced tools. We employ a hard negative mining approach, coupling rule-based procedure, GPT and zero-shot voice cloning for data synthesis, enriching both KWS and SID sub-processes. This strategy enhances our training effectiveness, ensuring our methods are well-equipped to handle the complexities of hard cases for KWS and open-set SID scenarios.

Our main contributions are as follows:

- Developed an innovative multi-modal, two-branch enrollment architecture for Query-by-Example KWS, optimizing phonetic and semantic feature extraction from input text to effectively merge text and audio modalities.
- Introduced an optimized training approach for open-set Speaker Identification in household scenarios, utilizing the proposed SRPL algorithm. This approach focuses on rapid, few-shot learning for unknown-speaker aware open-set SID, leveraging the WavLM-Xvector speaker encoder for enhanced performance.
- Implemented a comprehensive data augmentation strategy that integrates advanced tools, including zero-shot voice cloning, rule-based procedures, and Large-Language Model (GPT)-based negative sample mining, to support the construction of the PUKWS pipeline.
- Conducted individual evaluations of the proposed M-UDKWS and SRPL algorithms, and a pipeline evaluation of the proposed PUKWS, using thorough metrics to demonstrate the effectiveness of our approaches.

2. Related Work

2.1. User-defined Keyword Spotting

User-defined keyword spotting (UDKWS) aims to detect whether the target words appear in consecutive speech. There are mainly two types of existing UDKWS.

The traditional approach utilizes a large vocabulary continuous speech recognition (LVCSR), followed by a keyword spotting module that searches for keywords in the lattice generated by the LVCSR module [29, 30]. The LVCSR module uses a large number of audio-text pairs to train a traditional automatic speech recognition model on the generated lattice, containing decoded information for a given speech. This approach provides high accuracy and enables easy keyword customization without retraining. However, it is inefficient due to redundant information and experiences significant performance decline when handling out-of-vocabulary words [1, 5].

To address the inefficiency of LVCSR, researchers have focused on the Query-by-Example (QbyE) approach [16, 17], which consists of two main approaches: Query-by-Audio (QbyA) and Query-by-Text (QbyT). The QbyA approach utilizes acoustic features extracted from enrolled audio and query audio to measure their similarity using an efficient matching algorithm. [1] used frame-level bottleneck feature (BNF) combined with the dynamic time warping (DTW) matching algorithm. [5] improved the performance of UDKWS by using phoneme posterior probability (PPP) as a second-stage KWS model for multistage detection. Acoustic word embedding (AWE) [18, 19, 20], which maps variable-length speech signals into fixed-dimension word embeddings and utilizes metric algorithms such as cosine similarity to replace the inefficient DTW, has also found wide application. [31] extended the method to acoustic span embedding to achieve better results in detecting phrases.

To enhance the model's robustness to speech variations, recent research efforts have focused on QbyT. [21] exploits the correspondence between text and speech, outperform traditional QbyA approaches. Expanding upon this finding, [22, 23] further enhances the encoder and matching methods, significantly improving QbyT performance. [24] proposes the use of memory to implement the conversion of text features to speech features, reducing the mismatch between text and speech feature spaces and enhancing the model's robustness against homophones through the generation of confusing words. Conversely, [25] utilizes a larger scale pre-training approach, thereby improving the effectiveness of QbyT.

2.2. Speaker Identification and Open-set Learning

Speaker identification encompasses two main scenarios: closed-set and open-set. In closed-set speaker identification, test utterances are assumed to originate from a pre-enrolled speaker. This scenario typically employs multi-class classification loss for tuning [14, 32], prototype learning loss to bolster few-shot learning capabilities [33, 34], or graph-based learning methods [27]. However, these approaches primarily focus on optimizing closed-set classification, which may not fully address real-world application needs.

Recent efforts in the machine learning field are increasingly directed towards enabling open-set learning capabilities, as seen in studies on Reciprocal Points Learning or Adversarial Reciprocal Points Learning [28, 35, 36]. These methods have shown promising results in enhancing open-set recognition across various computer vision tasks. Open-set speaker identification,

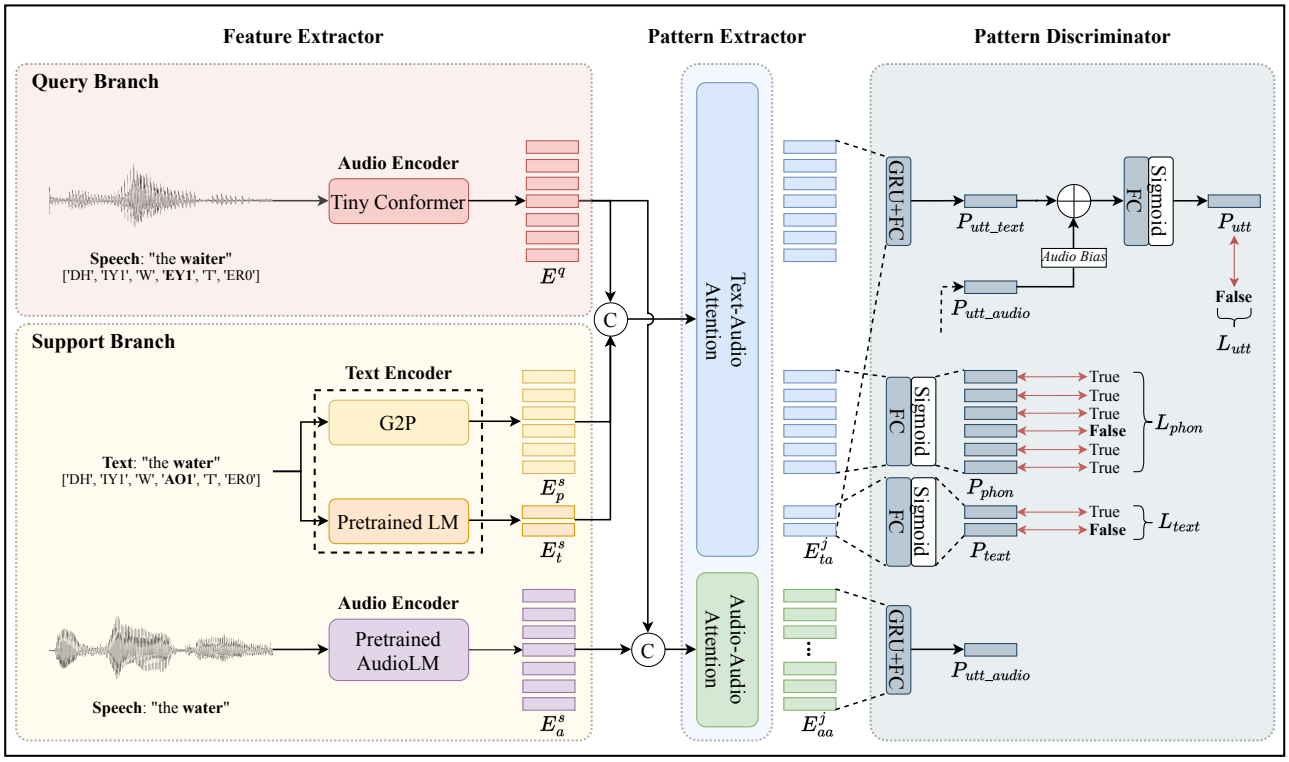


Figure 2: *Overview of the M-UDKWS framework.*

wherein the test utterance might be from an unenrolled "guest" speaker, is more aligned with real-world situations where not all utterances originate from known speakers. Although these approaches using i-vector based systems [37], or prototype-based loss learning [38, 13, 39], have demonstrated certain advancements, they often fall short of optimal performance in complex real-world scenarios.

In the realm of speaker recognition, recent advancements have prominently featured the development of advanced speaker encoders, such as X-vectors [40], ResNet [13], and the state-of-the-art WavLM based pretrained-audio models [41] or other self-supervised learning (SSL) based audio models [42]. These encoders mark significant technological progress in the field. However, despite the utilization of these advanced speaker encoders, the quest to refine and optimize Speaker Identification (SID) algorithms remains critical. This underscores the necessity for continued exploration and enhancement to fully leverage these advancements in addressing the complex challenges of open-set speaker identification.

3. Multi-Modal User-Defined Keyword Spotting (M-UDKWS)

In this section, we introduce our proposed method, the Multi-Modal User-defined Keyword Spotting (M-UDKWS), as illustrated in Figure 2. The M-UDKWS framework comprises three integral processes: feature extractor, pattern extractor, and pattern discriminator. The M-UDKWS model distinctively employs both keyword text and audio as multi-modal templates in its support branch, significantly enhancing its ability to accurately discriminate whether the features extracted from query audio are aligned with these support templates. This approach is crucial in facilitating dual-modality enrollment during the inference phase.

3.1. Model architecture

3.1.1. Feature Extractor

The feature extractor module consists of a support branch and a query branch, as shown in Figure 2. Inspired by EMKWS [23] and CED [24], we use the Conformer [43] architecture as the ***query branch*** of the audio encoder to generate the embedding for a given audio input signal. The Conformer integrates the powerful properties of self-attention to learn global interactions and convolutions to effectively capture local correlations. In the keyword spotting task, the lightweight Conformer performs well [23, 24].

For the ***support branch***, we employ a novel approach using both keyword text and template audio as multi-modal targets. The encoding of support keyword texts begins with the application of a pre-trained *Grapheme-to-Phoneme (G2P)* model [21, 22, 24] for capturing phonetic features. Additionally, *DistilBERT*[1] [23, 44], a model adept at natural language understanding, is used to enhance word-level encoding. This approach, integrating both phonetic and semantic representations, yields a richer embedding of the input text. Furthermore, we define template audio as an adaptive bias for bridging the gap between keyword text space and query audio space. For this, we utilize the pre-trained *WavLM*[2][41], which excels in generating comprehensive audio features. WavLM's ability to create effective acoustic embeddings provides a deeper insight into the nature of audio signals.

[1] https://huggingface.co/distilbert-base-uncased
[2] https://github.com/microsoft/unilm/tree/master/wavlm

Through the feature extractor module, we represent the ***query audio*** features as $E^q \in \mathbb{R}^{T^q \times d}$, the ***support phoneme*** features as $E_p^s \in \mathbb{R}^{T_p^s \times d}$, the ***support text*** features $E_t^s \in \mathbb{R}^{T_t^s \times d}$ and the ***support audio*** features as $E_a^s \in \mathbb{R}^{T_a^s \times d}$, where T^q represents the query audio frame length, T_p^s and T_t^s respectively represent the number of support phonemes and support subwords, T_a^s represents the length of the support audio frame, and d represents the frame dimension.

3.1.2. Pattern Extractor

The pattern extractor is based on the self-attention mechanism. As detailed in [22], the self-attention method does not require other modules in the fusion process of multiple modalities, presenting a concise and parameter-efficient approach for multimodal feature fusion. The matrix of attention outputs for a set of queries Q, with keys K and values V packed into matrices, is computed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

As demonstrated in Figure 2, the pattern extractor consists of a Text-Audio Attention module (TAA) and an Audio-Audio Attention module (AAA).

The ***TAA*** module processes three inputs: the query audio features $\mathbf{E^q}$, the support phoneme features $\mathbf{E_p^s}$, and the support text features $\mathbf{E_t^s}$. Its function is to conduct cross-modal matching between text and audio modalities, specifically determining if the query audio feature aligns with the target keyword text representation. To differentiate features derived from these sources, we use three learnable coding vectors e^{type}, each indicating the source of the features. Temporal position encoding, denoted as e^{pos} and following sinusoidal position encoding, is also incorporated. This results in the transformer inputs as shown in Equation (2).

$$\overline{E} = E + e^{pos} + e^{type} \tag{2}$$

Subsequently, the ***TAA*** module is applied to identify cross-modal correlations between support texts, support phonemes, and query audios. The transformed features $\overline{E^q}$, $\overline{E_p^s}$, and $\overline{E_t^s}$ are concatenated along the time dimension as E_{ta}^c, and the joint features $\mathbf{E_{ta}^j}$ is computed using self-attention as per Equation (1):

$$E_{ta}^c = (\overline{E^q}; \overline{E_p^s}; \overline{E_t^s}) \in \mathrm{R}^{(T^q+T_p^s+T_t^s)\times d} \tag{3}$$

$$E_{ta}^j = \text{Attention}(E_{ta}^c, E_{ta}^c, E_{ta}^c) \in \mathrm{R}^{(T^q+T_p^s+T_t^s)\times d} \tag{4}$$

The ***AAA*** module takes two specific inputs: the query audio embedding $\mathbf{E^q}$ and the support audio embedding $\mathbf{E_a^s}$. Its objective is to verify if the query audio features match any representation in the template audio. To distinguish between embeddings from these sources, we apply similar learnable type coding vectors and position encoding as outlined in Equation (2). The ***AAA*** module then processes these to identify matches within the audio modality. The joint features $\mathbf{E_{aa}^j}$ is calculated using the defined attention mechanism:

$$E_{aa}^c = (\overline{E^q}; \overline{E_a^s}) \in \mathrm{R}^{(T^q+T_a^s)\times d} \tag{5}$$

$$E_{aa}^j = \text{Attention}(E_{aa}^c, E_{aa}^c, E_{aa}^c) \in \mathrm{R}^{(T^q+T_a^s)\times d} \tag{6}$$

3.1.3. Pattern Discriminator

For the pattern discriminator, we utilize a GRU module to derive utterance-level posterior probabilities from the text-audio joint embedding $\mathbf{E_{ta}^j}$ and the audio-audio joint embedding $\mathbf{E_{aa}^j}$. These probabilities are denoted as $P_{utt_{text}}$ and $P_{utt_{audio}}$, and are computed as follows:

$$P_{utt\_text} = \text{GRU}(E_{ta}^j) \tag{7}$$

$$P_{utt\_audio} = \text{GRU}(E_{aa}^j) \tag{8}$$

Conceptually, the keyword spotting (KWS) decision primarily relies on the more stable output from 'audio query text', with 'audio query audio' providing supplementary information. Thus, we express the decision output as P_{utt_text}, enhanced by P_{utt_audio} to jointly assess utterance-level matching:

$$P_{utt} = \sigma(W^u \cdot (P_{utt\_text} + P_{utt\_audio}) + b^u) \tag{9}$$

Moreover, we incorporate text matching probabilities, P_{phon} and P_{text}, to improve frame-level matching between the query and support. As demonstrated in Figure 2, we separate the phoneme sequence $(\mathbf{E_{ta}^j})_{\mathbf{phon}}$ and the word sequence $(\mathbf{E_{ta}^j})_{\mathbf{text}}$ from the text-speech joint embedding $\mathbf{E_{ta}^j}$. The subscripts 'phon' and 'text' indicate frame indices in the ranges $(T^q, T^q + T_p^s]$ and $(T^q + T_p^s, T^q + T_p^s + T_t^s]$, respectively. These sequences are processed through a fully connected layer followed by a sigmoid function:

$$P_{phon} = \sigma(W^p \cdot (E_{ta}^j)_{phon} + b^p) \tag{10}$$

$$P_{text} = \sigma(W^t \cdot (E_{ta}^j)_{text} + b^t) \tag{11}$$

Here, W, b, and σ represent the trainable weights, biases, and sigmoid functions, respectively.

3.2. Training Approach

Our training objective is denoted $\mathcal{L}_{total}$, which consists of a combination of three binary cross-entropy (BCE) losses. These include the utterance-level loss ($\mathcal{L}_{utt}$) as the main loss, and two auxiliary losses: phoneme-level detection loss ($\mathcal{L}_{phon}$) and word-level detection loss ($\mathcal{L}_{text}$),

$$\mathcal{L}_{total} = \mathcal{L}_{utt} + \mathcal{L}_{phoneme} + \mathcal{L}_{text} \tag{12}$$

3.2.1. Utterance-level detection loss

The utterance-level detection loss is used to evaluate the similarity of the query branch and support branch. If the query audio is the target keyword, the label is 1, otherwise it is 0, and binary cross entropy (BCE) is used to calculate the loss.

3.2.2. Auxiliary detection loss

- **Phoneme-level detection loss**: We introduces phoneme-level detection loss to enhance the model's ability to distinguish between similar pronunciations (such as "the waiter" and "the water"). This process rely on alignment information of speech sounds and phonemes. 1 if the phoneme sequence of the speech tag matches the phoneme sequence of the keyword tag, 0 otherwise.

$$y_{phon} = \begin{cases} 1 & \text{if } y_{phon}^s = y_{phon}^q \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

Here, y_{phon}^s signifies the sequence of phonemes associated with the support keyword text, while y_{phon}^q represents the sequence of phonemes associated with the query keyword text.

- **Word-level detection loss**: To capitalize on the semantic differences between similar words and enhance the model's ability to discriminate, we introduce word-level detection loss. This approach is particularly effective for words like "waiter" and "water", where distinct text embeddings are evident. The process involves assigning a value of 1 when the text sequence of the query speech matches the text sequence of the target word. In cases where they do not align, a value of 0 is assigned.

$$y_{text} = \begin{cases} 1 & \text{if } y^s_{text} = y^q_{text} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

Here, y^s_{text} represents the word sequence associated with the support keyword text, and y^q_{text} represents the word sequence associated with the query keyword text.

4. Open-set Speaker Identification with Speaker Reciprocal Points

To refine embeddings for enhanced distinction within a household-specific domain, we propose employing lightweight adaptation models. These models adjust the speaker embeddings to better suit the target scenarios. As depicted in Figure 3, we augment the WavLM frontend with X-vector model to produce output E_{wv}, which, after transformation by the Lightweight Adaptor, yields the speaker-specific embedding $\mathbf{E_{spk}}$:

$$\mathbf{E_{spk}} = Light(Xvector(E_{wv})) \tag{15}$$

4.1. Training Approach with Speaker Reciprocal Points

Our innovative method for open-set Speaker Identification (SID) employs specialized learning techniques designed for accurate speaker recognition from limited data. Diverging from prototype learning loss methods [34, 39], our approach, inspired by [28], adopts Reciprocal Points Learning (RPL) and tailors it for SID, hence the term Speaker Reciprocal Points Learning (SRPL). This strategy is particularly suited for scenarios where the system encounters speakers not included in the training data. We apply a training approach that is aware of both known and unknown speakers, which is key to improving the system's accuracy and robustness for real-world application. As conceptually shown in Figure 4, this optimization ensures that known speaker embeddings are optimally distributed throughout the space, while concurrently securing a low magnitude area for the learning of unknown speakers

The WavLM X-vector front-end, optimized with an angle-based learning strategy, differs from the method in [28]. We measure the distance between reciprocal points (RP) and speaker embeddings through the inner product of the learnable RP and adapted feature embeddings. This inner product serves as a gauge for the proximity between embeddings and the class-specific learning RPs. SRPL strategically maximizes the distance between learnable embeddings and RPs:

$$\begin{aligned} d_d(\mathbf{E_{spk}}, P^k) &= \mathbf{E_{spk}} \cdot P^k, \\ d(\mathbf{E_{spk}}, P^k) &= -d_d(\mathbf{E_{spk}}, P^k). \end{aligned} \tag{16}$$

Class distances are employed to compute the optimization probabilities for known classes to enhance their distinctiveness:

$$p(y = k|\mathbf{E_{spk}}, P) = \frac{e^{d(\mathbf{E_{spk}}, P^k)}}{\sum_{i=1}^{N} e^{d(\mathbf{E_{spk}}, P^i)}}, \tag{17}$$

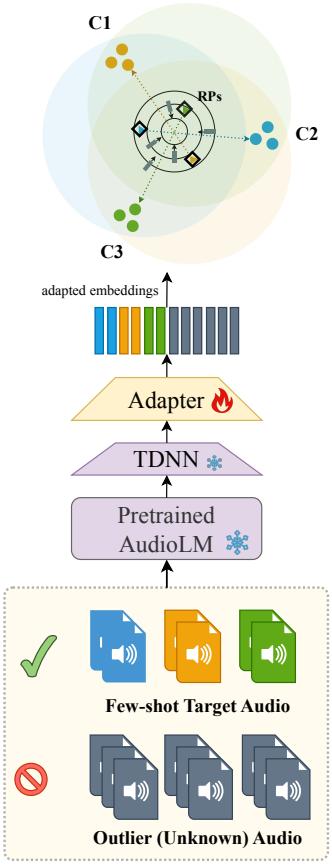


Figure 3: *Open-set SID with speaker reciprocal points learning (SRPL).*

$$\mathcal{L}_c(x; \theta, P) = -\log p(y = k|x, P), \tag{18}$$

where P^k are the RPs for the corresponding known speakers and P is defined as the set of all RPs.

The essence of RPL lies in the optimization of the speaker manifold, particularly in confining samples from a general pool of unknowns D_U within a predefined radius R. The primary aim, as outlined in [28], is to ensure that the maximum distance between sample sets D_U and $D_L^{\neq k}$ to the RPs P^k remains within R. Here, $D_L^{\neq k}$ refers to the negative samples known from class k.

$$\max(d(D_L^{\neq k} \cup D_U, P^k)) \leq R. \tag{19}$$

which is equally formulated as,

$$\begin{aligned} d_e(\mathbf{E_{spk}}, P^k) &= \| \mathbf{E_{spk}} - P^k \|_2^2, \\ \mathcal{L}_o(x_i; \theta, P^k, R) &= \max(d_e(\mathbf{E_{spk}}, P^k) - R, 0). \end{aligned} \tag{20}$$

The Euclidean distance $d_e(\cdot)$ is leveraged to ensure the magnitude optimization for unknowns is effectively contained. This objective is articulated as optimizing the embeddings in relation to the reciprocal points, which act as benchmarks to limit the space occupied by unknowns:

$$\mathcal{L}_{RPL}(\mathbf{E_{spk}}, y; \theta, P, R) = \mathcal{L}_c(\mathbf{E_{spk}}; \theta, P) + \lambda \mathcal{L}_o(\mathbf{E_{spk}}; \theta, P, R). \tag{21}$$

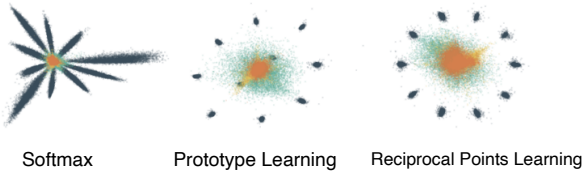


Figure 4: *Open-set Learning Loss Comparison, adapted from [45].*

4.2. Adversarial Enhancement with Negative Audio Instance

It is important to note that for typical household few-shot SID tasks, the acquisition of D_U is not usually considered, meaning that Equation (21) is optimized with known samples from D_L alone. However, negative samples representing unknowns can be sourced either through data augmentation, as discussed in Section 5, or collected randomly in real-world settings. To effectively utilize negative samples, [28] suggests using a Generative Adversarial Network (GAN) to create Confusion Samples (CS), which has shown to improve results when trained with reciprocal points. For SID tasks, we refine this method to utilize audio-level confusion training samples and incorporate them into the SRPL learning framework. This integration involves using confusion data in the adversarial learning process to optimize the negative samples $z \in D_U$ and maximize their entropy $H(\cdot)$ across all RPs, thus achieving the SRPL-CS optimization goal:

$$\min_{\theta} \left[\mathcal{L}_{RPL}(\mathbf{E_{spk}}, y; \theta) - \beta \cdot H(z, P)\right], \tag{22}$$

5. Data Augmentation with Advanced Tools

To bolster the performance of the Multi-Modal User-Defined Keyword Spotting (M-UDKWS) system, particularly in challenging scenarios, and to enhance the performance of Open-Set Speaker Identification in detecting unknown speakers, we have developed a data augmentation pipeline, as illustrated in Figure 5. This pipeline consists of two main components: Negative Text Mining (NTM) and Voice Cloning (VC). Each component plays a crucial role in augmenting data for the keyword spotting (KWS) and speaker identification (SID) processes according to their specific requirements. For KWS, the NTM module generates ***homophones***, ***synonyms*** and ***permutation*** of the target keywords or phrases to facilitate mining difficult cases. Meanwhile, for the SID task, negative samples produced by this module are utilized to improve the system's capability in identifying unknown speakers.

5.1. Rule-Based NTM

The rule-based NTM is employed to expand the negative examples for KWS systems using predefined rules. To find ***homophones***, we employ a pre-trained Grapheme-to-Phoneme (G2P) model, transforming the subwords of training phrases and common words into phoneme sequences. We then calculate the phonetic distance between the subwords of the target and each word in our lexicon. The top 25 words with the shortest phonetic distances are selected to create phonetically similar alternatives to the target subwords. For the ***synonyms*** process, we harness a pre-trained FastText[3] model to derive word embeddings and determine the semantic similarity between the subwords of the target and each lexicon word. Here too, we select the 25 most semantically similar words to generate alternatives related to the target subwords. These procedures employ a standard English word dictionary[4] containing approximately 10,000 entries.

The subsequent step involves substituting subwords in the target phrase to generate variants that are phonetically or semantically close to the original, thereby producing potentially confusing phrases.

Additionally, the ***permutation*** process, which shuffles the order of subwords in the target phrase, creates variations with identical phonemes but arranged differently, introducing complexity and enhancing the model's robustness.

5.2. LLM-Based NTM

To enhance the variety of challenging cases further, we utilize interactions with Generative Pre-trained Transformer (GPT) models. This strategy, as depicted in Figure 5, generates hard samples that closely mimic semantic or phonetic properties of the target words or phrases. These samples are more aligned with the natural language patterns encountered in everyday communication.

5.3. Voice Cloning Process

In the KWS workflow, after applying NTM, we introduce a zero-shot Voice Cloning (VC) module to create speech samples that are independent of speaker identity. This is achieved for each negative example by varying the speakers and speech rates. Utilizing this approach, the M-UDKWS system constructs a comprehensive dataset with over 170,000 negative examples.

For SID, our data augmentation concentrates on producing a substantial number of negative or confusion samples to simulate unknown speakers, as outlined in Section 4.2. We generate these samples using the VC module, with reference speakers chosen at random from the LibriTTS dataset. It is ensured that the voiceprints of these speakers do not overlap with those in subsequent evaluations. This data supports the strengthening of the SRPL process and facilitates the implementation of the SRPL-CS algorithm.

We deploy the TSCM-TTS system[5], a zero-shot voice cloning model trained on the LibriTTS corpus for both KWS and SID processes.

6. Experiments

In this section, we describe the experimental setup, including the datasets, the evaluation metrics, the pre-trained models used, and the implementation details of training and inference.

6.1. Datasets

- **LibriPhrase:** We constructed the Libriphrase training and test datasets using LibriSpeech [46], following methodologies from [21, 22, 23, 24]. The training dataset was generated from train-clean-100/360, and the test dataset from train-others-500. Libriphrase test dataset comprises two subsets: ***Libriphrase Easy (LE)*** and ***Libriphrase Hard (LH)***. Our model's performance is evaluated on these subsets, focusing on binary classification accuracy.

[3] https://fasttext.cc
[4] https://github.com/cmusphinx/cmudict
[5] https://great-research.github.io/tsct-tts-demo

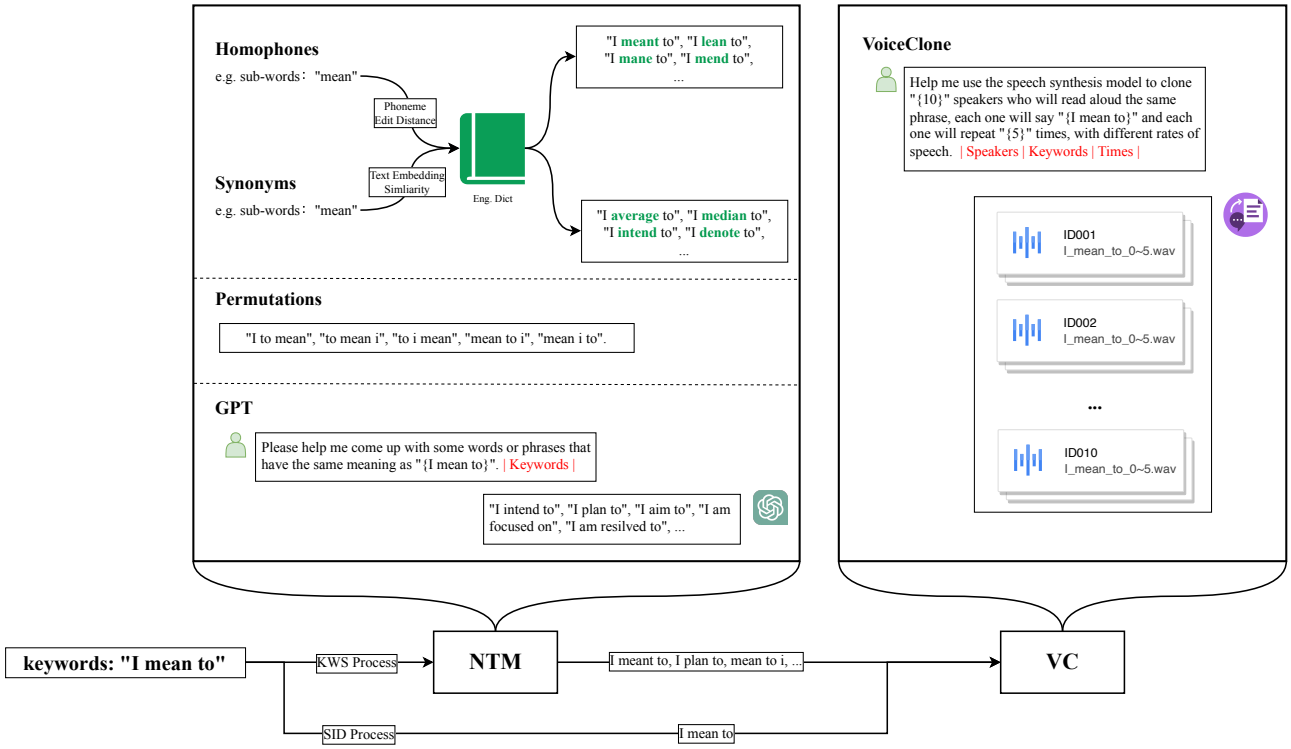


Figure 5: *Data augmentation process with advanced tools*

- **Google Speech Commands:** The Google Speech Commands dataset [47], a prevalent corpus for keyword spotting with 30 keywords, is employed for evaluation. We focus on 10 short command words, set up as a binary classification task with query audio and support text, abbreviated as ***G***. Additionally, we adopt a "one-vs-many" multi-class classification setting, named ***G-Many***, where 10 target keywords are randomly selected, and the remaining 20 keywords serve as unknown classes, following the configuration of [48].
- **Qualcomm Keywords Dataset:** This dataset, abbreviated as ***Q***, consisting of 4,270 utterances of four English keywords spoken by 50 speakers, is ideal for evaluating both KWS and SID tasks within our personalized user-defined keyword spotting pipeline. It also serves to assess the performance of our proposed open-set SID method in a household multi-class classification context.
- **Hey Snips:** The Hey Snips dataset is a speaker-independent collection of wake words with approximately 11K keyword and 86.5K non-keyword utterances. We adhere to the setup in [49] to evaluate one-word spotting or wake word detection performance, measuring the error rate over a specified duration.
- **AVSP-0:** To challenge our open-set SID method in more complex scenarios, we introduce the AVSP dataset, a multi-speaker, text-independent collection. AVSP-0 is a unique 'in the wild' dataset, comprising cross-age speakers and gathered from various online resources. Detailed information about AVSP-0 is provided in Appendix A.2.

6.2. Metrics

6.2.1. *For KWS Subtask Evaluation*

For the KWS sub-task, we employ the Equal Error Rate (EER) and the Area Under the Receiver Operating Characteristic Curve (AUC) as principal metrics, as detailed in [21, 22]. For the Google Speech Command dataset, we further assess multiclass classification performance using Acc(close) and Acc(open), as per [48], where Acc(close) excludes unknown class instances.

For the Qualcomm Keywords and Hey Snips datasets, which simulate practical wake-up word detection scenarios, we evaluate False Reject Rate (FRR) metrics at extremely low False Alarm rates, specifically at 1 False Alarm per hour (FAH=1) and 0.5 False Alarm per hour (FAH=0.5), and also consider scenarios with no False Alarms (FAH=0.05).

6.2.2. *For SID Subtask Evaluation*

The SID task follows the multi-class classification metrics process described in [28]. AUC is employed as a threshold-independent metric, plotting the true positive rate against the false positive rate, indicating the likelihood that a positive example is scored higher than a negative one. However, AUC does not account for known class accuracy in open-set recognition. Hence, we adopt the Open Set Classification Rate (OSCR) [50, 28] for evaluating open-set SID:

$$CCR(\delta) = \frac{|\{x \in D_T^k \mid \arg\max_k P(k|x) = \hat{k} \wedge P(\hat{k}|x) \geq \delta\}|}{|D_T^k|}, \tag{23}$$

$$FPR(\delta) = \frac{|\{x \mid x \in D_U \wedge \max_k P(k|x) \geq \delta\}|}{|D_U|}, \tag{24}$$

Here, δ is a threshold that requires thorough evaluation, and OSCR is defined as the area under the curve of the Correct Classification Rate (CCR) for known classes against the False Positive Rate (FPR) for unknown data.

6.2.3. For PUKWS Pipeline Evaluation

For personalized keyword spotting pipeline assessment, we consider metrics suitable for household personalized keyword detection scenarios. We introduce Keyword Aware AUC (K-AUC) and Keyword Aware Open-set Recognition Rate (K-OSCR) to reflect accuracy in both KWS and open-set SID:

$$\begin{aligned} KAUC &= FRR_{kws} * AUC_{sid}, \\ KOSCR &= FRR_{kws} * OSCR_{sid}. \end{aligned} \tag{25}$$

Given that FRR_{kws} represents the False Reject Rate of the KWS system at nearly no False Alarm Rate (FAH=0.05), it can be inferred that all phrases passed to the SID system are indeed the target keywords. Therefore, we denote the FRR_{kws} is considered the initial accuracy loss of the KWS, compounded with the subsequent stage metrics AUC_{sid} and $OSCR_{sid}$, to form a composite metric for the entire PUKWS pipeline evaluation.

6.3. Training and Inference Details

6.3.1. Training Details

- ***Training M-UDKWS:*** We expand the Libriphrase dataset using the data augmentation pipeline detailed in Section 5 and proceed to train the M-UDKWS system end-to-end. we extract 80-dimensional features for the audio signals in the query branch, which are then extracted into 128-dimensional frame6-level audio embeddings after tiny conformer. In the support branch, we encode the text into 128-dimensional phoneme embeddings using G2P and 768-dimensional word embeddings encoded by the text encoder DistilBERT, as well as encode the template audio into 768-dimensional frame-level audio embeddings using the WavLM module, respectively. It is worth noting that all supported modules are fixed-parameter. Subsequently, three lightweight mappers were used to convert all inputs into a uniform 128-dimensional space. The training process uses the Adam optimizer with about 50k steps of training. Specific model parameters are detailed in Appendix A.3.
- ***Few-shot Fine-tuning of M-UDKWS:*** To adapt M-UDKWS for specific wake words on the Qualcomm and Hey Snips datasets, we apply few-shot fine-tuning. This involves constructing a large number of challenging negative examples for each target keyword using the data augmentation pipeline from Section 5, alongside a small number of real-world keyword utterances (e.g., 5, 10, and 50). The pre-trained M-UDKWS model undergoes fine-tuning with the techniques described in Section 3.1, utilizing the Adam optimizer at a reduced learning rate over 5k steps.
- ***Few-shot Fine-tuning of SRPL for Open-set SID:*** For the open-set SID system, we rapidly fine-tune using SRPL or SRPL-CS for few-shot scenarios. In the enrollment phase, 5 to 15 speakers are designated as ***known*** speakers, with 20 utterances each for model tuning. When applying SRPL without confusion samples, only known samples are used for training. For SRPL with confusion samples, 1000 utterances from ***unknown*** speakers are generated as negative samples as detailed in Section 5.3 or gathered from unused part of the Qualcomm Keywords dataset. The adaptation involves a 3-layer MLP model, and linearly transforms to K-way speaker outputs. The SRPL loss, as outlined in Section 4.1, is then applied. SGD optimizer at a small learning rate is used to only fine-tune the adapter for 1k steps.

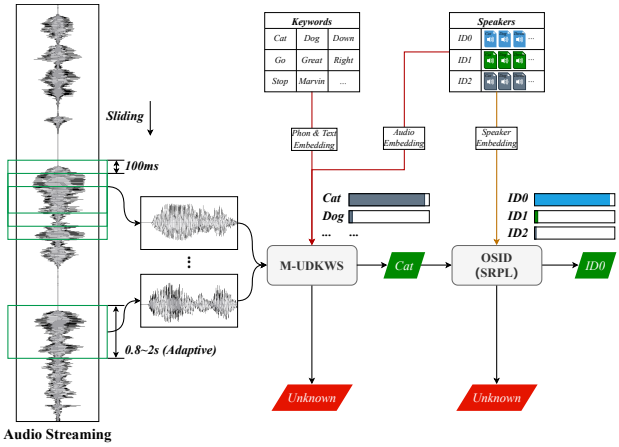


Figure 6: *Detailed Inference Process of the PUKWS Pipelines.*

6.3.2. Inference Details

The inference process of our system unfolds in two main stages. During the ***enrollment phase***, we enroll the target keywords and template audio for the user. This step involves extracting phoneme and text embeddings, as well as audio embeddings. For the SRPL-based open-set SID system, this phase includes quick fine-tuning of the speaker adapter using the known target speakers' template audio. In the subsequent ***testing phase***, the M-UDKWS system searches for keywords. The detection output is the utterance-level probability score P_{utt}, as depicted in Figure 2. For the SRPL-based open-set SID system, we forgo using cosine similarity [41] and instead compute the probability score by measuring the logits of the test speaker embeddings against the reciprocal points, resulting in the speaker identification probability as defined in Equation (17).

Figure 6 illustrates the household PUKWS's practical application. The M-UDKWS system is always on, monitoring the speech stream for keywords within a specified detection window (e.g., 1 second for short words, 2 seconds for longer phrases), and evaluating scores at 100ms intervals. The open-set SID system activates only upon detection of a keyword. If a target speaker is identified, the keyword and speaker identity are simultaneously relayed to the next system level to facilitate additional tasks for the AI agent.

7. Results

7.1. Comparative Evaluation of M-UDKWS

We compare the proposed M-UDKWS model with the SOTA in Table 1. Triplet [51], Attention [52], and DONUT [53] use the QbyA approach, and CMCD [21], EMKWS [23], CED [24], and PhonMatchNet [22] are the recently proposed QbyT approach. AdaKWS [25] uses large-scale pre-training with fine-tuning on the LibriPhrase training set, which has a large number of parameters.

Evaluation results show that among all the methods without confusing keyword generation, our method achieves higher area under the ROC curve (AUC) and lower equal error rate (EER), which is better than all the QbyA and QbyT methods. Com-

Method	# Params	EER(%) ↓				AUC(%) ↑			
		G	Q	LE	LH	G	Q	LE	LH
w/o confusable keyword generation									
Triplet [51]	-	35.60	38.72	32.75	44.36	71.48	66.44	63.53	54.88
Attention [52]	-	14.75	49.13	28.74	41.95	92.09	50.13	78.74	62.65
DONUT [53]	-	31.65	18.23	14.67	35.22	66.36	89.69	92.29	69.58
CMCD [21]	0.65M	27.25	12.15	8.42	32.90	81.06	94.51	96.70	73.5
EMKWS [23]	3.70M	-	-	7.36	23.36	-	-	97.83	84.21
CED [24]	3.60M	14.05	-	0.80	18.40	93.16	-	99.94	89.20
PhonMatchNet [22]	0.65M	6.77	4.75	2.80	18.82	98.11	98.90	99.29	88.52
M-UDKWS (ours)	3.50M	5.17	3.05	0.52	13.63	98.93	99.29	99.97	93.06
M-UDKWS-TA† (ours)	3.90M	**4.85**	**1.52**	**0.47**	**13.03**	**98.96**	**99.89**	**99.98**	**93.73**
w/ confusable keyword generation									
CED [24]	3.60M	13.45	-	1.70	14.40	93.94	-	99.84	92.70
AdaKWS-Tiny [25]	15M	-	-	1.61	13.47	-	-	99.80	93.75
AdaKWS-Small [25]	109M	-	-	1.21	11.48	-	-	99.82	95.09
M-UDKWS (ours)	3.50M	5.52	2.31	0.78	10.57	98.69	99.71	99.94	95.37
M-UDKWS-TA† (ours)	3.90M	**4.86**	**2.15**	**0.69**	**9.55**	**98.97**	**99.74**	**99.95**	**96.08**

Table 1: *Experimental results of M-UDKWS methods in various datasets. G: Google Commands V1, Q: Qualcomm Keyword Speech dataset, LE: LibriPhrase-Easy, LH: LibriPhrase-Hard. (†): our model with enrollment audio.*

Method	Open		Close	Open		Close
	AUC(%)↑	OSCR(%)↑	ACC(%)↑	AUC(%)↑	OSCR(%)↑	ACC(%)↑
	5Way			10Way		
WavLM X-vector [41]	84.09	83.61	99.54	83.81	81.49	96.06
+ Softmax [32]	71.33	70.95	99.07	66.46	66.91	98.91
+ ProtoType [33, 34]	71.54	71.39	99.54	88.87	88.45	98.91
+ OpenFEAT [39]	71.54	71.39	99.54	88.87	88.45	98.91
+ SRPL	81.12	80.95	99.53	89.53	89.16	98.47
+ SRPL-CS(F)	92.51	92.48	99.90	92.06	91.25	98.69
+ SRPL-CS(R)	**95.73**	**94.25**	99.21	**95.40**	**94.26**	98.81

Table 2: *Experimental results of open-set SID tasks on Qualcomm dataset.*

pared with PhonMatchNet, M-UDKWS improves the EER of LE and LH by 2.28% and 5.19%, respectively. M-UDKWS-TA with support audio achieves further improvement in all metrics.

M-UDKWS with confusable keyword generation in the system has a relative improvement of 3.06% and 2.31% on EER and AUC of LH. M-UDKWS-TA with template audio as support audio further improves on the EER and AUC of LH by 9.55% and 96.08% respectively, significantly outperforming all methods. The performance of M-UDKWS-TA with a parameter count of 3.9M is even better than that of AdaKWS with a parameter count of 109M, which shows the excellence of our approach.

7.2. Comparative Evaluation of SRPL

Table 2 presents the comparative results of our SRPL system against various baseline methodologies. The WavLM X-vector system, which utilizes cosine similarity for speaker identification, serves as a fundamental baseline. This system simplifies the SID task to a "one-vs-one" binary classification problem without specialized optimization for multi-class scenarios. It computes the probability of each target speaker based on the cosine similarity between the test embeddings and the mean of the enrollments.

Advanced methods, such as Softmax fine-tuning [32], Prototype learning [33, 34], and OpenFEAT learning [39], are tailored for known speakers within specific household settings. They may or may not include unknown speaker detection and use logits for scoring. Our SRPL system outperforms both the Prototype and OpenFEAT methods, as well as the WavLM X-vector baseline. This superiority is evident in the evaluation metrics of AUC and OSCR, where SRPL demonstrates robust performance for both 5-way and 10-way classification tasks.

Incorporating real confusion samples during training, the SRPL-CS(R) achieves remarkably high OSCR scores of 94.25% and 94.26% in 5-way and 10-way evaluations, respectively, marking significant advancements over the baselines. Similar efficacy is observed with synthetic confusion samples created via the zero-shot VC system detailed in Section 5, where SRPL-CS(F) also attains outstanding results. Additionally, the auxiliary close-set metric ACC, which focuses solely on the recognition accuracy of close-set known speakers, indicates that the SRPL-CS system maintain perfect close-set performance.

Method		K-AUC%↑	K-OSCR%↑
KWS	SID		
Baseline System			
PhonMatchNet [22]	WavLM X-vector [41]	57.6	58.9
PhonMatchNet	+ Softmax [32]	51.2	50.0
PhonMatchNet	+ OpenFEAT [39]	51.2	50.3
Ours			
M-UDKWS-ZS	+ SRPL-CS(R)	82.0	82.0
M-UDKWS-FS	+ SRPL-CS(R)	92.1	91.2
M-UDKWS-TA-FS	+ SRPL-CS(R)	**93.5**	**93.7**

Table 3: *Overall Process for the PUKWS Piplines Evaluations.*

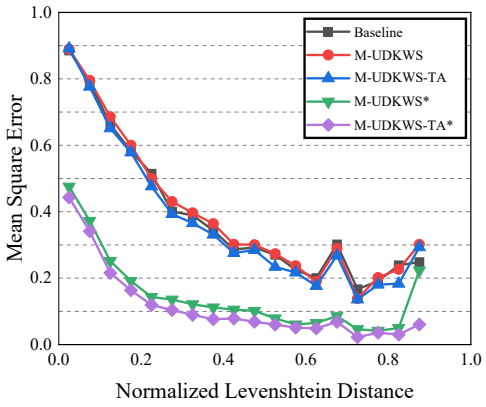


Figure 7: *Performance Analysis via Keyword Similarity on Libriphrase Hard (LH) Dataset. Models marked with an asterisk (*) are enhanced with confusable keyword generation.*

7.3. Comparative Performance Evaluation of PUKWS Pipeline Systems

Table 3 demostrate the comparative performance of our complete PUKWS pipeline against established baselines, particularly highlighting the state-of-the-art User-defined KWS system, PhonMatchNet, coupled with the WavLM X-vector and optimized open-set SID systems for this task. The efficacy of our integrated PUKWS pipeline, which combines M-UDKWS with SRPL-CS, is evident when measured against the baselines using the K-AUC and K-OSCR metrics. Notably, on the Qualcomm Keywords dataset, our M-UDKWS-TA with SRPL-CS method achieves a K-OSCR of 93.7%, markedly outperforming the baseline systems by 34.8%.

7.4. Supplementary Investigations and Analyses

In this section, we delve into additional explorations and analyses that supplement our core findings.

7.4.1. Performance Analysis via Keyword Similarity

We investigated the impact of hard cases on model performance by utilizing the normalized Levenshtein distance [22] to quantify the similarity of hard negative keywords at the phoneme level, where closer distances signify greater difficulty in differentiation. We computed the Mean Square Error (MSE) between the hard negative samples' model predictions and the true labels (which are zeros), constraining the MSE values within the range of 0 to 1.

Figure 7 illustrates the evaluation results for Libriphrase Hard (LH) cases. It reveals that the model's accuracy diminishes with the increasing difficulty of cases. In comparison, the baseline model exhibits poor performance, particularly with confusable words that are harder to distinguish. By incorporating confusable keywords for auxiliary training, our model demonstrates a marked improvement in detecting hard cases. Furthermore, the model that leverages both audio and text inputs for support consistently outperforms the one relying solely on text. This enhancement is primarily attributed to the audio input's role in diminishing the disparity between text and audio spaces.

Method	# of supports	G-Many	
		Acc(close)↑	Acc(open)↑
QbyA [48]	1	69.0±1.67	66.0±1.03
	5	90.5±0.53	80.6±0.44
M-UDKWS (ours)	0	93.9±0.15	88.2±0.31
	1	95.4±0.14	90.7±0.18
	5	**95.9+0.14**	**90.8±0.18**

Table 4: *The zero-shot performance of M-UDKWS in multi-class classification evaluation "G-Many".*

Method	Hey Snips		Q
	FRR@FAH=0.5↓	@FAH=1↓	@FAH=0.05↓
C-RIL [54]	3.53	2.82	-
WeKws* [49]	-	0.87	-
PhonMatchNet [22]	-	-	29.48
M-UDKWS-ZS	17.12	12.09	12.82
M-UDKWS-FS	**3.40**	**2.33**	3.20
M-UDKWS-TA-FS	-	-	**0.64**

Table 5: *M-UDKWS for Customizing Wake Word Performance on Hey Snips and Qualcomm.*

7.4.2. The Zero-Shot Performance of M-UDKWS in Multi-Class Classification Tasks

We assessed the performance of M-UDKWS in multi-class classification tasks using the G-Many dataset. This evaluation was particularly aimed at comparing M-UDKWS's effectiveness with the QbyA system [48] in a multi-classification context. Table 4 illustrates these findings. Notably, the QbyA system exhibits suboptimal performance with a limited number of enrollment audios. In contrast, M-UDKWS shows superior performance with just text input as support and continues to improve as users provide more enrollment audio.

7.4.3. M-UDKWS for customized wake word wake-up tasks

We evaluated our M-UDKWS method on the Hey Snips and Qualcomm datasets, with results summarized in Table 5. The initial zero-shot performance of M-UDKWS was not ideal when evaluated against this metric. Notably, all baseline models compared, including C-RIL [54] and WeKws [49], were trained with a substantial amount of data specific to these keywords ("full-shot"), which should be considered as a performance benchmark for zero-shot systems.

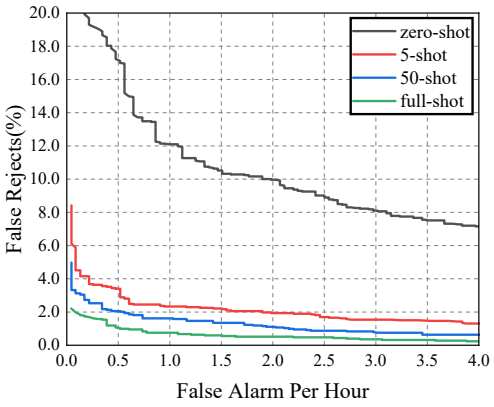


Figure 8: *Performance evaluation of M-UDKWS on the Hey Snips dataset.*

Method	EER(%) ↓		AUC(%)↑	
	LE	LH	LE	LH
M-UDKWS-TA	0.69	**9.55**	99.95	**96.08**
w/o confusable keywords	**0.47**	13.03	**99.98**	93.73
w/o support audio	0.52	13.63	99.97	93.06
w/o auxilary loss	0.63	15.29	99.97	91.91

Table 6: *Ablation studies of M-UDKWS.*

Consequently, we applied the 5-shot fine-tuning method described in Section 6, leading to significant improvements in M-UDKWS's performance, to the point of being comparable with the baseline keyword-specific systems. The performance further enhanced when enrolled audio was utilized. Additionally, fine-tuning experiments on the Hey Snips dataset with an increasing number of fine-tuning audios indicated that system performance improved proportionally, peaking with a corpus of 50-shots, as detailed in Figure 8.

7.4.4. Visualization of attention map of the text support branch

We analyzed the attention map of our Text-Audio Attention (TAA) model, as shown in Figure 9. Our observations revealed that the attention map tends to exhibit significant monotonicity when aligned with the target word. This pattern suggests that the TAA module effectively aligns speech segments with the target word, highlighting areas of significant relevance at specific positions. This is observable both at the word level, where attention relates to Language Model (LM) features E_t^s, and at the phoneme level, with attention guided by Grapheme-to-Phoneme (G2P) features E_p^s, as illustrated in Figure 2. Notably, individual subwords elicit distinct responses.

For confusable negative samples, matched subwords or phonemes exhibit high levels of brightness on the attention map, indicating active engagement, whereas non-matching segments display lower brightness levels. The model prediction score inversely correlates with the monotonicity of the detection: lower scores are associated with non-monotonic detections, whereas perfect monotonic matches result in higher scores.

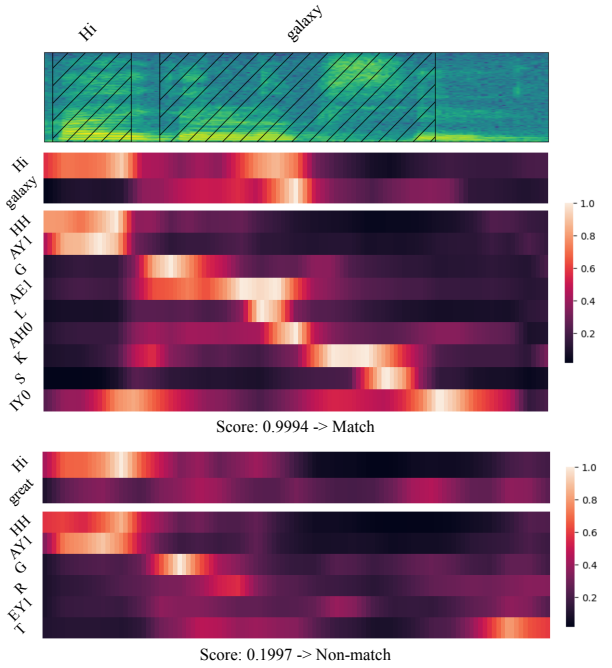


Figure 9: *Overview of the data compos.*

7.4.5. Ablation studies of M-UDKWS

Table 6 presents our ablation study results, indicating that the integration of confusable keyword generation, the support audio branch, and auxiliary loss all contribute to the enhanced performance of our model. Specifically, the inclusion of auxiliary loss and the support audio branch yields performance gains across both Libriphrase Easy (LE) and Libriphrase Hard (LH) datasets. While the generation of confusable keywords slightly reduces performance on LE, it notably enhances the model's effectiveness on LH. This disparity suggests a trade-off, where enhancing the model's ability to distinguish hard cases may impact performance on simpler tasks.

7.4.6. Auxiliary Studies for SRPL for Open-set SID

Figure 10 illustrates the speaker manifold embeddings. Upon examining the open-set distributions of these embeddings, it becomes evident that for the WavLM baseline methods, the embedding spaces are largely indistinguishable from the unknowns. The baseline model without open-set SID rapid optimization exhibiting reduced speaker discriminability. Conversely, SRPL rapid training based on WavLM Xvector demonstrates superior performance in discriminating speaker embeddings and effectively segregating and clustering unknowns into separate positions, while the embedding of the known target speakers are better clustered. This highlights the efficacy of the proposed open-set learning objective.

Table 7 further assesses the SRPL approach using the AVSP-0 dataset—a text-independent, "in the wild" speaker dataset featuring cross-age variability, as detailed in Section A.2. The results underscore the robustness of SRPL methods, showcasing their superior performance across baselines in complex and challenging speaker identification scenarios.

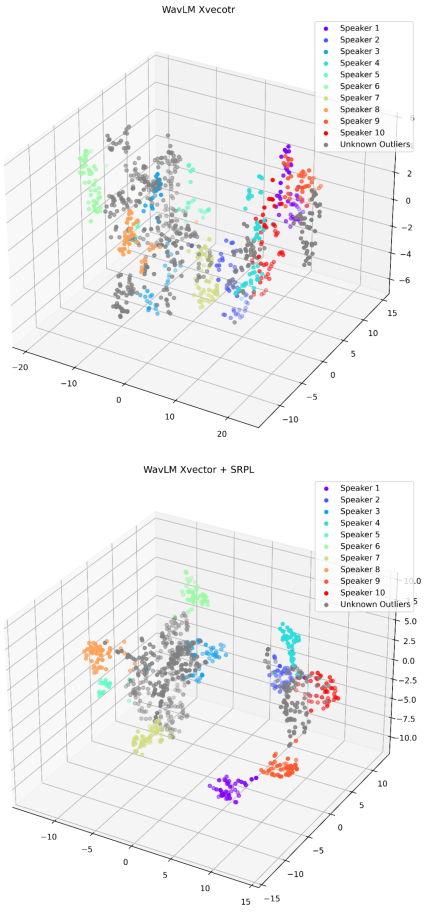


Figure 10: *TSNE Visualization of the embedding for open-set SID optimization.*

8. Conclusions

In conclusion, this paper introduces a personalized user-defined keyword spotting (PUKWS) pipeline optimized for household environments. We propose a multi-modal user-defined keyword spotting (M-UDKWS) approach and a speaker reciprocal points learning (SRPL) algorithm for open-set speaker identification. By utilizing data augmentation strategies with advanced tools, we achieve a significant performance improvement of PUKWS, which is 34.8% compared to the existing baseline. Our research contribution lies in the development of an innovative pipeline that combines the functionality of user-defined keyword spotting and open-set speaker identification, providing significant improvements to the utility and user experience of smart household environments. In the future, we plan to further extend our approach, including testing it in more real scenarios and exploring related research such as lightweight deployment of the model to further improve the performance and robustness of the PUKWS.

9. Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants 61871262, 61901251, and 62071284, the Innovation Program of Shanghai Municipal Science and Technology Commission under Grants 21ZR1422400, 20JC1416400 and 20511106603, Pudong New Area Science & Technology Development Fund, Key-Area Research and Development Program of Guangdong Province under Grant 2020B0101130012, and Foshan Science and Technology Innovation Team Project under Grant FS0AA-KJ919-4402-0060. We would like to express our gratitude to the various funding agencies that have supported this work.

Method	Openset		Closeset
	AUC(%)↑	OSCR(%)↑	ACC(%)↑
	15Way		
WavLM X-vector [41]	76.0	66.0	79.8
+ Softmax [32]	82.3	76.0	87.0
+ ProtoType [33, 34]	81.6	73.9	84.6
+ SRPL	**85.6**	**77.3**	**86.1**
	5Way		
WavLM X-vector	84.7	80.9	91.6
+ Softmax	83.9	81.0	94.0
+ ProtoType	81.6	73.9	84.6
+ SRPL	**86.3**	**82.8**	**93.2**

Table 7: *Experimental results of open-set SID tasks on AVSP-0 dataset*

10. References

[1] J. Wang, Y. He, C. Zhao, Q. Shao, W.-W. Tu, T. Ko, H. yi Lee, and L. Xie, "Auto-KWS 2021 Challenge: Task, Datasets, and Baselines," in *Proc. Interspeech 2021*, 2021, pp. 4244–4248.

[2] Y. Jia, X. Wang, X. Qin, Y. Zhang, X. Wang, J. Wang, D. Zhang, and M. Li, "The 2020 Personalized Voice Trigger Challenge: Open Datasets, Evaluation Metrics, Baseline System and Results," in *Proc. Interspeech 2021*, 2021, pp. 4239–4243.

[3] D. Liao, J. Li, Y. Zhi, S. Li, Q. Hong, and L. Li, "An Integrated Framework for Two-Pass Personalized Voice Trigger," in *Proc. Interspeech 2021*, 2021, pp. 4633–4637.

[4] J. Hou, L. Zhang, Y. Fu, Q. Wang, Z. Yang, Q. Shao, and L. Xie, "The npu system for the 2020 personalized voice trigger challenge," *arXiv preprint arXiv:2102.13552*, 2021.

[5] Y. Wang, Y. Jia, M. Ma, Z. Cai, and M. Li, "The dku system description for the interspeech 2021 auto-kws challenge," *arXiv preprint arXiv:2104.04993*, 2021.

[6] J. Wang, M. Xu, J. Hou, B. Zhang, X.-L. Zhang, L. Xie, and F. Pan, "Wekws: A production first small-footprint end-to-end keyword spotting toolkit," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[7] H. Wang, C. Liang, S. Wang, Z. Chen, B. Zhang, X. Xiang, Y. Deng, and Y. Qian, "Wespeaker: A research and production oriented speaker embedding learning toolkit," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[8] E. Variani, X. Lei, E. McDermott, I. L. Moreno, and J. Gonzalez-Dominguez, "Deep neural networks for small footprint text-dependent speaker verification," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014. [Online]. Available: http://dx.doi.org/10.1109/icassp.2014.6854363

[9] R. Rikhye, Q. Wang, Q. Liang, Y. He, D. Zhao, Y. Huang, A. Narayanan, and I. McGraw, "Personalized Keyphrase Detection Using Speaker and Environment Information," in *Proc. Interspeech 2021*, 2021, pp. 4204–4208.

[10] S. Yang, B. Kim, I. Chung, and S. Chang, "Personalized Keyword Spotting through Multi-task Learning," in *Proc. Interspeech 2022*, 2022, pp. 1881–1885.

[11] A. Q. Ohi, M. F. Mridha, M. A. Hamid, and M. M. Monowar, "Deep speaker recognition: Process, progress, and challenges," *IEEE Access*, vol. 9, pp. 89 619–89 643, 2021.

[12] Z. Bai and X.-L. Zhang, "Speaker recognition based on deep learning: An overview," *Neural Networks*, vol. 140, pp. 65–99, 2021.

[13] R. Peri, S. O. Sadjadi, and D. Garcia-Romero, "Voxwatch: An open-set speaker recognition benchmark on voxceleb," *arXiv preprint arXiv:2307.00169*, 2023.

[14] R. Li, J.-Y. Jiang, C. Wu, C.-C. Hsieh, and A. Stolcke, "Speaker identification for household scenarios with self-attention and adversarial training," 2020.

[15] Z. Tan, Y. Yang, E. Han, and A. Stolcke, "Improving speaker identification for shared devices by adapting embeddings to speaker subsets," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 1124–1131.

[16] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5236–5240.

[17] B. Kim, M. Lee, J. Lee, Y. Kim, and K. Hwang, "Query-by-example on-device keyword spotting," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 532–538.

[18] H. Kamper, W. Wang, and K. Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 4950–4954.

[19] N. Sacchi, A. Nanchen, M. Jaggi, and M. Cernak, "Open-vocabulary keyword spotting with audio and text embeddings," in *INTERSPEECH 2019-IEEE International Conference on Acoustics, Speech, and Signal Processing*, no. CONF, 2019.

[20] K. Audhkhasi, A. Rosenberg, A. Sethy, B. Ramabhadran, and B. Kingsbury, "End-to-end asr-free keyword search from speech," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1351–1359, 2017.

[21] H. Shin, H. Han, D. Kim, S. Chung, and H. Kang, "Learning audio-text agreement for open-vocabulary keyword spotting," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pp. 1871–1875, 2022.

[22] Y.-H. Lee and N. Cho, "PhonMatchNet: Phoneme-Guided Zero-Shot Keyword Spotting for User-Defined Keywords," in *Proc. INTERSPEECH 2023*, 2023, pp. 3964–3968.

[23] K. Nishu, M. Cho, and D. Naik, "Matching latent encoding for audio-text based keyword spotting," in *Interspeech*, 2023.

[24] K. Nishu, M. Cho, P. Dixon, and D. Naik, "Flexible keyword spotting based on homogeneous audio-text embedding," *arXiv preprint arXiv:2308.06472*, 2023.

[25] A. Navon, A. Shamsian, N. Glazer, G. Hetz, and J. Keshet, "Open-vocabulary keyword-spotting with adaptive instance normalization," *ArXiv*, vol. abs/2309.08561, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:262012802

[26] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *ArXiv*, vol. abs/1910.01108, 2019.

[27] F. Tong, S. Zheng, M. Zhang, Y. Chen, H. Suo, Q. Hong, and L. Li, "Graph convolutional network based semi-supervised learning on multi-speaker meeting data," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 6622–6626.

[28] G. Chen, P. Peng, X. Wang, and Y. Tian, "Adversarial reciprocal points learning for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 8065–8081, 2021.

[29] G. Chen, S. Khudanpur, D. Povey, J. Trmal, D. Yarowsky, and O. Yilmaz, "Quantifying the value of pronunciation lexicons for keyword search in lowresource languages," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 8560–8564.

[30] P. Motlicek, F. Valente, and I. Szoke, "Improving acoustic based keyword spotting using lvcsr lattices," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4413–4416.

[31] Y. Hu, S. Settle, and K. Livescu, "Acoustic span embeddings for multilingual query-by-example search," in *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2021, pp. 935–942.

[32] Q.-B. Hong, C.-H. Wu, H.-M. Wang, and C.-L. Huang, "Combining deep embeddings of acoustic and articulatory features for speaker identification," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7589–7593.

[33] J. Wang, K.-C. Wang, M. T. Law, F. Rudzicz, and M. Brudno, "Centroid-based deep metric learning for speaker recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3652–3656.

[34] Y. Li, H. Chen, W. Cao, Q. Huang, and Q. He, "Few-shot speaker identification using lightweight prototypical network with feature grouping and interaction," *IEEE Transactions on Multimedia*, 2023.

[35] G. Chen, L. Qiao, Y. Shi, P. Peng, J. Li, T. Huang, S. Pu, and Y. Tian, "Learning open set network with discriminative reciprocal points," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 507–522.

[36] H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Robust classification with convolutional prototype learning," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3474–3482.

[37] K. Wilkinghoff, "On open-set speaker identification with i-vectors." in *Odyssey*, 2020, pp. 408–414.

[38] J. Fortuna, P. Sivakumaran, A. Ariyaeeinia, and A. Malegaonkar, "Open-set speaker identification using adapted gaussian mixture models," in *Ninth European conference on speech communication and technology*, 2005.

[39] K. Kishan, Z. Tan, L. Chen, M. Jin, E. Han, A. Stolcke, and C. Lee, "Openfeat: Improving speaker identification by open-set few-shot embedding adaptation with transformer," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7062–7066.

[40] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[41] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.

[42] J.-w. Jung, W. Zhang, J. Shi, Z. Aldeneh, T. Higuchi, B.-J. Theobald, A. H. Abdelaziz, and S. Watanabe, "Espnet-spk: full pipeline speaker embedding toolkit with reproducible recipes, self-supervised front-ends, and off-the-shelf models," *arXiv preprint arXiv:2401.17230*, 2024.

[43] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[44] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[45] A. Berg, M. O'Connor, and M. T. Cruz, "Keyword transformer: A self-attention model for keyword spotting," *arXiv preprint arXiv:2104.00769*, 2021.

[46] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[47] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.

[48] D. Lee, M. Kim, S. H. Mun, M. H. Han, and N. S. Kim, "Fully unsupervised training of few-shot keyword spotting," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023, pp. 266–272.

[49] J. Wang, M. Xu, J. Hou, B. Zhang, X.-L. Zhang, L. Xie, and F. Pan, "Wekws: A production first small-footprint end-to-end keyword spotting toolkit," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[50] A. R. Dhamija, M. Günther, and T. Boult, "Reducing network agnostophobia," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

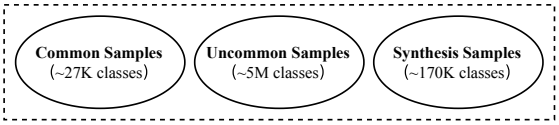
[51] N. Sacchi, A. Nanchen, M. Jaggi, and M. Cernak, "Open-vocabulary keyword spotting with audio and text embeddings," in *INTERSPEECH 2019-IEEE International Conference on Acoustics, Speech, and Signal Processing*, no. CONF, 2019.

[52] J. Huang, W. Gharbieh, H. S. Shim, and E. Kim, "Query-by-example keyword spotting system using multi-head attention and soft-triple loss," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6858–6862.

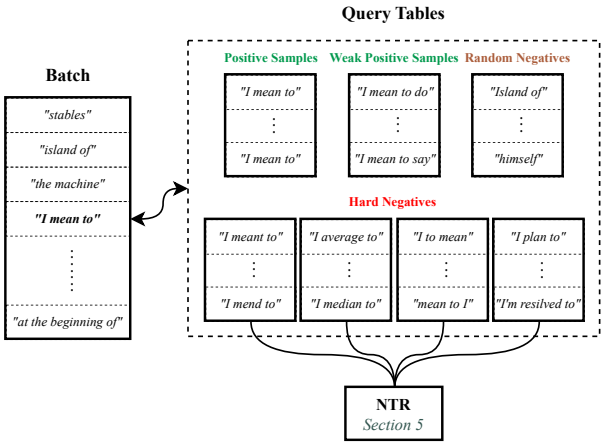
[53] L. Lugosch, S. Myer, and V. S. Tomar, "Donut: Ctc-based query-by-example keyword spotting," *arXiv preprint arXiv:1811.10736*, 2018.

[54] K. Zhang, Z. Wu, D. Yuan, J. Luan, J. Jia, H. Meng, and B. Song, "Re-Weighted Interval Loss for Handling Data Imbalance Problem of End-to-End Keyword Spotting," in *Proc. Interspeech 2020*, 2020, pp. 2567–2571.

[55] J. Liao, H. Duan, K. Feng, W. Zhao, Y. Yang, and L. Chen, "A light weight model for active speaker detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 932–22 941.



(a) Data composition



(b) Batching scheme

Figure 11: *Overview of M-UDKWS Data Composition and Batching Scheme.*

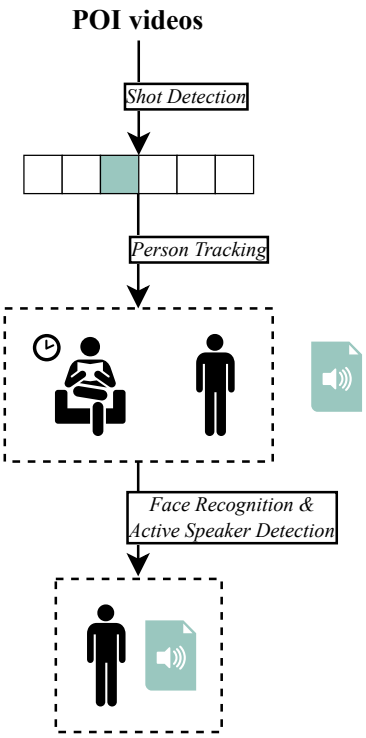


Figure 12: *The AVSP dataset collection pipeline.*

A. Appendices

A.1. M-UDKWS Data Composition and Batching Scheme

The M-UDKWS data composition and batch processing scheme is summarized in Figure 11. In the training phase, we categorized the Libriphrase into common and uncommon samples based on the criterion of whether the number of audios within a category is greater than 20. Meanwhile, with the data enhancement scheme shown in Figure 5, we synthesized a large number of samples for the common samples, totaling about 27K keyword categories. The number of uncommon samples is about 5M keyword categories, while the synthesized samples generated by speech synthesis are about 170K keyword categories. This comprehensive data processing and enhancement strategy helps to improve the performance of the model in various scenarios and ensures its generalization ability for different categories and samples.

The batching scheme for M-UDKWS is shown in Figure 11(b), where we construct a mini-batch for the target word in each batch, and the samples are randomly sampled from the query table in a 1:1 ratio of positive and negative samples. The query table describes the composition of our positive and negative samples, which mainly includes positive samples, weak positive samples, random negatives, and hard negatives. Where weak positive samples are phrases obtained from the dataset that contain sequences of positive samples, and hard negatives are obtained from the data augmentation pipeline shown in Figure 5 that includes hard cases such as synonyms, homophones, etc.

A.2. AVSP0 Dataset

AVSP is a cross-age speaker dataset that we collected using the data collection pipeline shown in Figure 12, containing speech data from nearly 300 speakers with an average voice age span of 8 years. In this paper, we select 50 speakers from AVSP, with vast amount of audio data for each speaker, to simulate the speaker identification task in a home environment, which we name AVSP0. The data collection pipeline is briefly described below:

- **STEP1: Candidate list of POIs.** AVSP data aims to collect massive audiovisual data of target speakers with a long age span for aging and personalization research. We selected about 300 famous American TV series actors[6], YouTube[7]and Bilibili[8] video bloggers as POIs through manual screening.
- **STEP2: Shot Detection.** In order to avoid scene switching from affecting the identification of the speaker, we use the ffmpeg tool[9] to detect scene changes and crop the video into multiple clips.
- **STEP3: Person Tracking.** For each shot, we use YoloV8[10] for person tracking, splitting each tracked ID into separate video sub-segments, which is faster and more efficient than conventional face detection and tracking.
- **STEP4: Face Recognition and Active Speaker Detection.** We use face recognition[11] and active speaker detection [55] to determine the correspondence between the audio track and the sub-video based on the segmented sub-video clips and the corresponding audio track.

[6]https://www.imdb.com
[7]https://www.youtube.com
[8]https://www.bilibili.com
[9]https://ffmpeg.org
[10]https://docs.ultralytics.com
[11]https://github.com/deepinsight/insightface

Hyper-parameter				Params
M-UDKWS				
Feature Extractor	Conformer	Encoder Layers	6	2.9M
		Attention Heads	4	
	G2P			0.83M
	DistilBERT			66M
	WavLM			94.7M
Pattern Extractor	Text-Audio Attention	Encoder Layers	2	0.4M
		Attention Heads	4	
	Text-Audio Attention	Encoder Layers	2	0.4M
		Attention Heads	4	
Pattern Discriminator	GRU & Linear			0.2M
Total Number of M-UDKWS Parameters				3.9M
Open-set SID				
Feature Extractor	WavLM			94.7M
Feature Adaptor	Linear			0.3M

Table 8: *Model configurations*

A.3. Model Configurations

We list the hyper-parameters of M-UDKWS and open-set SID in Table 8. We utilize pre-trained G2P, DistilBERT, and WavLM for M-UDKWS. These models are used for offline extraction of enrolled keyword features, which are not reused during inference, and thus do not re-consume arithmetic, which needs only a small amount of memory. For open-set SID, this module is not always on. Generally, it needs to be woken up through M-UDKWS detection and then activate the open-set SID module. We use the WavLM module as the open-set SID feature extraction frontend. When finetuning, only the added Adapter module is training.