

SELF-SUPERVISED PRETRAINING FOR ROBUST PERSONALIZED VOICE ACTIVITY DETECTION IN ADVERSE CONDITIONS

Holger Severin Bovbjerg¹, Jesper Jensen^{1,2}, Jan Østergaard¹, Zheng-Hua Tan^{1,3}

¹Department of Electronic Systems, Aalborg University, Denmark

²Oticon A/S, Denmark

³Pioneer Centre for AI, Denmark

{hsbo, jje, jo, zt}@es.aau.dk

ABSTRACT

In this paper, we propose the use of self-supervised pretraining on a large unlabelled data set to improve the performance of a personalized voice activity detection (VAD) model in adverse conditions. We pretrain a long short-term memory (LSTM)-encoder using the autoregressive predictive coding (APC) framework and fine-tune it for personalized VAD. We also propose a denoising variant of APC, with the goal of improving the robustness of personalized VAD. The trained models are systematically evaluated on both clean speech and speech contaminated by various types of noise at different SNR-levels and compared to a purely supervised model. Our experiments show that self-supervised pretraining not only improves performance in clean conditions, but also yields models which are more robust to adverse conditions compared to purely supervised learning.

Index Terms— Self-Supervised Learning, Voice Activity Detection, Target Speaker, Deep Learning

1. INTRODUCTION

Being able to detect the presence of speech in a potentially noisy signal is a commonly utilized processing step in modern speech processing systems, generally referred to as voice activity detection (VAD). A VAD system has to classify whether a short frame of audio contains speech, usually from speech features like Mel-filterbank features, in an unsupervised [1] or supervised manner [2]. Applications include using a VAD model as a preprocessing step for automatic speech recognition, as a gating mechanism for microphones in online meeting devices, or as a part of a speech enhancement system. For real-time applications, such as speech enhancement for hearing aids, it is desired that the VAD model is low-latency and low complexity, while also being robust in adverse conditions such as background noise.

Recently, personalized VAD models, which are also able to determine whether the speech is from a target speaker, have been proposed [3, 4, 5, 6]. These personalized VAD models introduce a number of interesting capabilities, such as removing false-triggering on background speakers, at the expense of also needing to model speaker characteristics. Training a model to detect voice activity and distinguish between target speech and non-target speech in a supervised fashion requires a large amount of speech data from several speakers annotated with both VAD labels and framewise speaker identity. For example, [3] and [4] utilize 960 h of annotated speech from 2338 different speakers to train their models, while [6] utilize up to 27 500 h of annotated speech. Although high-quality VAD labels can be automatically obtained using forced-alignment [7], rel-

atively clean speech and a corresponding transcript of the speech signal is required. Additionally, frame-level speaker identity labels can be difficult to obtain. Therefore, the adoption of Personalized VAD models is limited by the ability to obtain such large labelled data sets, restricting their widespread adoption.

Self-supervised learning (SSL) methods provide a means of utilizing unlabelled data, which is easier to obtain in large quantities. Models pretrained using SSL have shown state-of-the-art performance in many domains, including speech processing [8, 9, 10], and have also been shown to learn more robust features than purely supervised models [11, 12]. The application of SSL for pretraining of VAD models is currently unexplored, although one study used speech features from a pretrained wav2vec2 [8] model as input features to a VAD model, and found that they perform better than standard Mel-spectrogram features [13]. However, using a large pretrained speech model (95M+ parameters) for feature extraction arguably defeats the purpose of having a small, efficient VAD model.

In this work, we propose to use a simple SSL framework known as Autoregressive Predictive Coding (APC), to directly pretrain a small LSTM-encoder, with the aim of improving performance and robustness in adverse conditions, when fine-tuning for personalized VAD. Additionally, we propose a denoising variant of APC for improved robustness. Here, we modify the APC framework to predict future clean speech frames from noisy input features, as opposed to predicting clean future frames from clean input features.

We carry out experiments, using both clean and noisy training data generated through online multistyle training (MTR) [14]. The trained models are evaluated systematically on both clean test data and on different test sets containing either seen or unseen noise at SNR levels ranging from -5 dB to 20 dB. The results show the following:

1. APC pretraining and fine-tuning on clean data, leads to an absolute improvement in mean average precision (mAP) of 1.9% compared to supervised training, when evaluating the models in clean conditions. Interestingly, an average absolute improvement of 6.05% is observed for noisy conditions, while not having seen any noise during training.
2. When using MTR, APC pretraining leads to an average absolute improvement for seen noise 4.8% over the baseline, while a further absolute improvement 2.3% is observed for the proposed DenoisingAPC. For unseen noise, DenoisingAPC+MTR also achieves the best performance, with an absolute improvement of 8% compared to baseline+MTR.

The source code used to produce the results of this paper is made publicly available.¹

2. METHODOLOGY AND DATA SETS

This section describes our personalized VAD model, the APC pre-training framework and the data used for training and testing.

2.1. Personalized VAD model

Our personalized VAD model is inspired by the Personal VAD system presented in [3] and is illustrated in Figure 1. The personalized VAD classifies input log Mel-filterbank features as either non-speech (ns), target-speaker speech (tss) or non-target-speaker speech (ntss). We choose to separate the speaker verification and VAD tasks into separate modules and reside to using an already trained model for speaker verification. More specifically, we focus on training a robust VAD model, and use a separate already trained d-vector model [15] to extract speaker embeddings for speaker verification.

The VAD module predicts the probability of speech or no speech, z^s and z^{ns} , for any given input frame. The d-vector model generates an embedding which is compared to the target-speaker embedding through cosine similarity to generate a target-speaker similarity score s . As s is a similarity score, its value might not necessarily represent the probability of a target-speaker being present. Therefore, s is scaled by learnable parameters α and β such that the scaled similarity becomes $s' = s\alpha + \beta$. Finally, the VAD output and scaled similarity score are combined such that

$$z_t^k = \begin{cases} z_t^{ns} & k = ns, \\ s' z_t^s & k = tss, \\ (1 - s') z_t^s & k = ntss. \end{cases} \quad (1)$$

where z_t^k is the output corresponding to class k at time frame t and is used to classify each frame as either non-speech (ns), target-speaker speech (tss) or non-target-speaker speech (ntss).

In [3], the authors also propose a personalized VAD where the target-speaker embedding is simply concatenated to the features, which is then used as input to the VAD. Here, the VAD model also learns to extract speaker characteristics through implicit knowledge distillation from the speaker embedding model, removing the need for a separate speaker embedding model during runtime. However, our implementation failed to achieve good performance using this approach.

For the speaker embedding model, we use a freely available d-vector model as described in [15]. The d-vector model used in this work, is pretrained on VoxCeleb [16] and LibriSpeech-other data. It has 3 LSTM layers, each with a hidden dimension of 256 producing 256-dimensional d-vector speaker embeddings and has a total of 1.4M parameters. As in [3], our VAD model is a 2-layer LSTM with a hidden dimension of 64, yielding a total of 60k parameters. Both the d-vector and VAD models take 40-dimensional log Mel-filterbank features as input, computed from 25 ms frames with frame shift of 10 ms.

Target-speakers are enrolled by generating a d-vector embedding using one or more enrolment utterances (minimum 5 s). Here, speaker embeddings are computed by sliding a 1.6 s window across the enrolment utterances with a shift of 0.4 s, generating an embedding for each window position, which are then averaged to generate the target-speaker d-vector embedding.

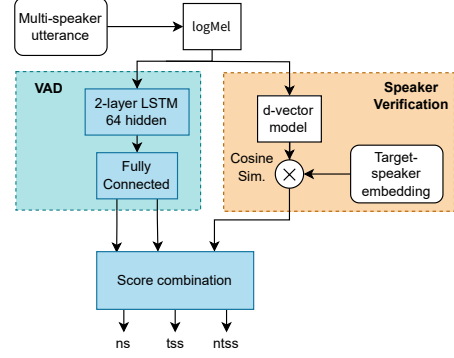


Fig. 1. Overview of the personalized VAD system model used in this work. We only train the blue boxes, while we use an already trained d-vector model with fixed weights.

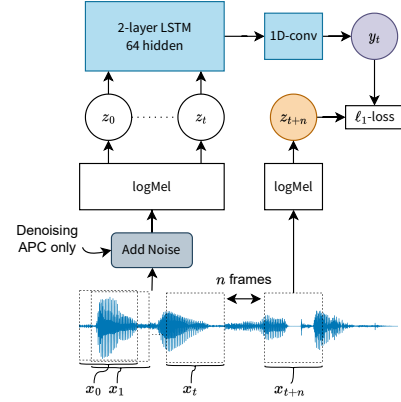


Fig. 2. Overview of the APC framework and DenoisingAPC variant. Note that noise is only added for DenoisingAPC, while this is skipped for standard APC.

2.2. Autoregressive predictive coding

Inspired by the success of pretraining language models, the Autoregressive Predictive Coding [17] framework predicts future speech features from the current and previous feature vectors. More formally, given a sequence of feature vectors z_0, \dots, z_t computed from frames x_0, \dots, x_t , we ideally seek a model f that predicts z_{t+n} such that $y_t = f(z_0, \dots, z_t) = z_{t+n}$, denoting z_t as the feature vector and y_t as the output at time frame t . APC thus builds on the notion that if a model is able to predict the future, it must have a good representation of the past and present. An overview of the APC framework, including the denoising variant, is illustrated in Figure 2. Models pretrained using APC have been shown to learn both speaker and content information and shown good performance for a number of downstream tasks [18].

While more complex SSL methods such as wav2vec2 [8] can also learn information from future frames, APC only encodes information from previous frames. Thus, the learned representation does not rely on future information. This is particularly suitable for VAD applied to real-time applications, as the VAD model is desired to be causal. In our experiments, we feed the log Mel-filterbank features to a 2-layer LSTM encoder, and use a single 1D-convolutional layer to project the hidden representations back to the input feature space.

¹<https://github.com/HolgerBovbjerg/SelfSupervisedPVAD>

In addition to the standard APC framework, we also propose a denoising variant of APC. Here, speech features are extracted from both clean speech and speech which have been corrupted by noise. We then predict future clean features from noisy input features, as depicted in Figure 2. This forces the model to extract information related to the source signal from a noisy mixture, thus learning to distinguish the source signal from background noise.

2.3. Data sets

As mentioned in [3], the amount of readily available multi-speaker data with natural speaker turns, as well as speaker identity information is limited, and as a result we carry out experiments on a simulated multi-speaker data set. While speakers might overlap in realistic settings, such as a cocktail party scenario, it has been found that a personalized VAD model trained on non-overlapping speech also performs well on overlapping speech [4]. Following [3], we uniformly sample 1 to 3 utterances from individual speakers and a target-speaker is randomly selected from one of the individual utterances. We then simply concatenate the utterances to generate multi-speaker utterances.

For our experiments, we use the freely available Librispeech [19] data set to construct both training and test data. Librispeech consists of 960 h training data split into two sets of 100 h and 360 h categorized as “clean”, and a 500 h “other” set, which is “less clean”. Similarly, both a “clean” and an “other” set is available for testing. Table 1 shows a summary of the different data sets used in our experiments for training, pretraining and testing.

In our experiments, we only use utterances within the same set when constructing multi-speaker utterances. For pretraining, the multi-speaker utterances constructed from the *train-clean-100*, *train-clean-360* and *train-other-500* sets are used, yielding a total of all 960 h speech. For supervised training, only multi-speaker utterances generated from the 100 h *train-clean-100* set are used. Additionally, we also train the models on the 10h LibriLight [20] training set. This simulates a setting with a large pool of unannotated data available for pretraining and a smaller pool of labelled data available for supervised training. As Librispeech includes speech transcripts, VAD labels are generated using forced-alignment [21], while we generate framewise speaker labels using the speaker identity information included in the Librispeech metadata.

For testing the model performance in clean conditions, we use the utterances generated from the *test-clean* set. To be able to evaluate the trained model in varying adverse conditions, noisy test data is generated by adding noise, to the *test-clean* multi-speaker utterances. Here, we pick two environmental noise types, namely bus and café, and two speech-like noise types, namely babble and speech-shaped noise, each representing a realistic adverse condition. For each noise type, noisy tests set have been generated by adding the noise type at a specific SNR level, ranging from -5 dB to 20 dB in steps of 5 dB, yielding a total of 24 noisy test sets.

Table 1. Summary of data sets with concatenated utterances used for training, pretraining and testing.

Set	#Multi-speaker Utt.	#Speakers	Used for
train-100-clean	14252	251	SSL + Supervised
train-360-clean	52068	921	SSL
train-500-other	74177	1166	SSL
LibriLight 10h	1364	44	Supervised
test-clean	1315	40	Testing

2.4. Supervised baseline

When training the personalized VAD model, the pretrained d-vector model weights are fixed, and we only update the VAD network and fully-connected network depicted in Figure 1. We reuse the hyperparameter choices in [3] using a cross-entropy loss and a batch size of 64, the ADAM optimizer with an initial learning rate of $5 \cdot 10^{-5}$, and a gradual reduction of the learning rate following a cosine annealing schedule.

2.5. Pretraining and fine-tuning

During pretraining, only the LSTM-encoder in the VAD network depicted in Figure 1 is pretrained, yielding a system as seen in Figure 2. In [22] it is found that predicting $n = 3$ frames ahead during APC pretraining leads to good downstream task performance, thus we adopt this choice and use an ℓ_1 -loss as the objective function. We pretrain the LSTM encoder for 10 epochs using a batch size of 32 and ADAM optimizer with an initial learning-rate of 0.01, and a cosine annealing learning rate schedule. After pretraining, the LSTM encoder weights are copied to a Personal VAD model which is then fine-tuned, using the same procedure as used for training the supervised baseline.

2.6. Multistyle training

A commonly used technique to improve model robustness is adding various noise to the training data. This technique is generally referred to as multistyle training (MTR) and has been shown to improve model robustness [14]. Therefore, we also carry out experiments where we apply online MTR. Here, we add noise from different adverse conditions as described in [23], namely babble, bus, pedestrian, street and speech-shaped noise. We include babble, bus and speech-shaped noise in both training data and test sets, while keeping café noise unseen during training. The noise is added at varying SNR levels in the range -5 dB to 20 dB. We also add room acoustics from recorded RIRs as used in [24]. When applying MTR, randomly sampled noise and room acoustics are added to the individual multi-speaker utterances, each with a probability of 50 %.

2.7. Metrics

To evaluate the performance of the trained model, we follow [3] and compute the average precision score for each class and use the mean average precision (mAP) score as our main evaluation metric. For a given class, average precision is computed as

$$AP = \sum_n (R_n - R_{n-1}) \cdot P_n \quad (2)$$

with P_n and R_n being the precision and recall at threshold n . To compute the mAP score, we compute the AP score for each class and take the mean.

3. RESULTS

In the following section, the results from our experiments are presented. First, we analyse how the trained models perform in clean conditions, followed by an in-depth analysis of how the trained models perform in various adverse environments. All models have been trained using five different random seeds.

Table 2. Results on clean test set (mAP, %) and 95 % confidence intervals. Baseline is a purely supervised model, while APC and DN-APC are SSL pretrained models which have been fine-tuned for personalized VAD. MTR denotes that multistyle training was used during supervised training.

Model	AP			mAP
	ns	tss	ntss	
Baseline	82.5±1.13	94.5±0.22	94.8±0.22	90.6±0.47
+MTR	79.8±0.75	94.1±0.19	93.6±0.31	89.2±0.28
APC	86.9±0.54	94.9±0.11	95.6±0.01	92.5±0.23
+MTR	81.9±0.88	94.4±0.09	94.2±0.38	90.1±0.30
DN-APC	88.0±0.57	94.7±0.16	96.1±0.08	92.9±0.14
+MTR	81.9±0.31	94.2±0.30	95.2±0.30	90.4±0.10

3.1. Clean conditions

In Table 2 the performance of the various models on the clean test set is presented. Here, ns, tss and ntss denotes no-speech, target-speaker speech and non-targets-speaker speech, respectively.

Comparing the models without MTR, Baseline and APC, the APC pretrained model shows an improvement in mAP of 2.1 % and DN-APC an improvement of 2.5 %. The DN-APC model scores highest of all models with an mAP of 92.9 %. Using MTR for model robustness usually comes at the cost of a performance drop in clean conditions [14]. As expected, we observe that the models using MTR perform slightly worse, although the DN-APC model performs comparable to the baseline without MTR. When using MTR, the DN-APC pretrained model performs best, with an overall improvement of 1.3 % compared to the supervised baseline using MTR.

3.2. Adverse conditions

We evaluate the performance of the trained models in various adverse conditions, including background noise consisting of bus, babble, and speech-shaped noise. Additionally, we also evaluate the performance on a noise type (café noise) unseen during training, to evaluate whether the robustness of the models generalize to an unseen noise type.

Table 3 presents the mAP scores and 95 % confidence intervals of the trained models when testing for seen noise at different SNR-levels are reported. While Table 3 shows summary scores averaged over all noise types, the general picture for each individual noise type is the same.

Looking Table 3, the pretrained models clearly outperform the supervised baseline models in seen noise. Interestingly, the APC model outperforms the baseline by 5.6 % on average when neither is using MTR, without having seen any noise during pretraining or supervised training. Using MTR leads to a further improvement, while DN-APC+MTR yields the best results, outperforming the baseline+MTR by 7.1 % on average and APC+MTR pretraining by 2.3 %. For Table 4, showing results when evaluating the models in unseen noise, a similar pattern is observed, with DN-APC+MTR showing the best results, outperforming the baseline+MTR by 8 % on average and APC+MTR by 3 %.

In Table 5 the average performance for models trained on LibriLight 10h training set is presented. Here, we observe that the pretrained models outperform the supervised baselines by an even larger margin, with an average improvement of 24.3 % for DN-APC+MTR compared to baseline+MTR.

Table 3. Performance in seen noise (mAP, %) and 95 % confidence intervals.

SNR	Baseline	Baseline + MTR	APC	APC +MTR	DN-APC + MTR
-5	63.0±1.51	67.1±0.81	65.9±2.22	72.3±1.48	74.1±0.99
0	63.8±2.32	71.0±1.25	67.3±2.99	76.9±1.62	80.0±0.52
5	64.9±2.44	75.2±1.48	69.0±2.89	80.9±0.89	84.3±0.43
10	66.3±1.72	79.1±1.73	71.8±1.78	84.1±0.44	86.9±0.35
15	68.7±1.60	82.5±1.23	76.6±2.79	86.5±0.43	88.2±0.29
20	73.1±1.67	84.9±1.19	82.4±3.24	88.0±0.47	89.1±0.32
Avg.	66.6±1.73	76.6±1.13	72.2±1.43	81.4±0.75	83.7±0.46

Table 4. Performance in unseen noise (mAP, %) and 95 % confidence intervals.

SNR	Baseline	Baseline +MTR	APC	APC +MTR	DN-APC +MTR
-5	63.3±1.05	63.7±0.91	66.0±1.21	68.3±1.92	72.8±0.46
0	64.0±1.44	68.1±1.20	67.9±1.05	74.5±2.24	79.5±0.40
5	65.2±1.56	73.5±1.14	70.8±1.33	80.2±1.69	84.0±0.31
10	67.3±1.36	78.7±0.96	75.3±2.42	84.3±0.91	86.6±0.26
15	71.0±1.82	82.9±0.88	80.9±3.14	86.9±0.55	88.1±0.27
20	77.3±1.19	85.7±0.74	86.1±2.30	88.3±0.44	89.1±0.34
Avg.	68.0±2.19	75.4±0.82	74.5±1.44	80.4±1.16	83.4±0.29

In summary, using APC pretraining improves performance substantially in noisy conditions, and additionally improves performance in noisy conditions. Our proposed DN-APC in combination with MTR achieves the best performance, with an average improvement of 7.1 % in seen noise and 8 % in unseen noise compared to baseline+MTR.

4. CONCLUSIONS

In this paper, we proposed the use of self-supervised pretraining to leverage unlabelled data for improving the robustness of a personalized VAD model in adverse conditions. For pretraining we used the APC framework, while we also proposed a Denoising variant of APC for improved robustness. We compared the pretrained models with a supervised baseline and tested their performance in both clean and adverse conditions with both seen and unseen noise at various SNR-levels.

Our results show a significant improvement in robustness to background noise when using APC pretraining. Both APC and our proposed Denoising APC outperform the baseline, while our proposed Denoising APC achieves the best performance. Overall, it can be concluded that self-supervised pretraining can improve the personalized VAD performance in both clean and noisy conditions.

Table 5. Avg. performance for models trained on 10h LibriLight training set in clean, seen and unseen noise (mAP, %) and 95 % confidence intervals.

Test set	Baseline	Baseline +MTR	APC	APC +MTR	DN-APC +MTR
Clean	57.2±6.96	55.6±6.47	83.8±1.89	82.5±2.46	83.2±1.91
Seen	54.9±2.37	55.0±2.43	63.7±2.05	68.9±2.07	77.9±1.57
Unseen	56.2±2.45	56.3±2.45	65.0±2.87	69.6±3.73	78.6±1.46

5. REFERENCES

- [1] Zheng-Hua Tan, Achintya kr. Sarkar, and Najim Dehak, “rvad: An unsupervised segment-based robust voice activity detection method,” *Computer Speech & Language*, vol. 59, pp. 1–21, 2020.
- [2] Heinrich Dinkel, Shuai Wang, Xuenan Xu, Mengyue Wu, and Kai Yu, “Voice activity detection in the wild: A data-driven approach using teacher-student training,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1542–1555, 2021.
- [3] Shaojin Ding, Quan Wang, Shuo-Yiin Chang, Li Wan, and Ignacio Lopez Moreno, “Personal VAD: Speaker-Conditioned Voice Activity Detection,” in *Proc. The Speaker and Language Recognition Workshop (Odyssey 2020)*, 2020, pp. 433–439.
- [4] Ivan Medennikov, Maxim Korenevsky, Tatiana Prisyach, Yuri Y. Khokhlov, Mariya Korenevskaya, Ivan Sorokin, Tatiana Timofeeva, Anton Mitrofanov, Andrei Andrusenko, Ivan Podluzhny, Aleksandr Laptev, and Aleksei Romanenko, “Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario,” in *Proc. Interspeech 2020*, 2020.
- [5] Maokui He, Desh Raj, Zili Huang, Jun Du, Zhuo Chen, and Shinji Watanabe, “Target-speaker voice activity detection with improved i-vector estimation for unknown number of speaker,” in *Proc. Interspeech 2021*, 2021, pp. 2523–2527.
- [6] Shaojin Ding, Rajeev Rikhye, Qiao Liang, Yanzhang He, Quan Wang, Arun Narayanan, Tom O’Malley, and Ian McGraw, “Personal VAD 2.0: Optimizing Personal Voice Activity Detection for On-Device Speech Recognition,” in *Proc. Interspeech 2022*, 2022, pp. 3744–3748.
- [7] Ivan Kraljevski, Zheng-Hua Tan, and Maria Paola Bissiri, “Comparison of forced-alignment speech recognition and humans for generating reference VAD,” in *Proc. Interspeech 2015*, 2015, pp. 2937–2941.
- [8] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in neural information processing systems*, vol. 33, pp. 12449–12460, 2020.
- [9] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdel rahman Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [10] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, et al., “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, pp. 1505–1518, 2021.
- [11] Hong Liu, Jeff Z. HaoChen, Adrien Gaidon, and Tengyu Ma, “Self-supervised learning is more robust to dataset imbalance,” in *International Conference on Learning Representations*, 2022.
- [12] Zeping Luo, Shiyu Wu, Cindy Weng, Mo Zhou, and Rong Ge, “Understanding the robustness of self-supervised learning through topic modeling,” in *International Conference on Learning Representations*, 2023.
- [13] Sina Alisamir, Fabien Ringeval, and François Portet, “Cross-domain voice activity detection with self-supervised representations,” *ArXiv*, vol. abs/2209.11061, 2022.
- [14] Rohit Prabhavalkar, Razi Alvaraz, Carolina Parada, Preetum Nakkiran, and Tara N. Sainath, “Automatic gain control and multi-style training for robust small-footprint keyword spotting with deep neural networks,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4704–4708.
- [15] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, “Generalized end-to-end loss for speaker verification,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4879–4883.
- [16] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Senior, “Voxceleb: Large-scale speaker verification in the wild,” *Computer Speech & Language*, vol. 60, pp. 101027, 2020.
- [17] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James R. Glass, “An unsupervised autoregressive model for speech representation learning,” in *Proc. Interspeech 2019*, pp. 146–150.
- [18] Gene-Ping Yang, Sung-Lin Yeh, Yu-An Chung, James Glass, and Hao Tang, “Autoregressive predictive coding: A comprehensive study,” *IEEE Journal of Selected Topics in Signal Processing*, pp. 1–12, 2022.
- [19] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.
- [20] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, “Libri-light: A benchmark for asr with limited or no supervision,” in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7669–7673.
- [21] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger, “Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi,” in *Proc. Interspeech 2017*, 2017, pp. 498–502.
- [22] Yu-An Chung and James Glass, “Generative pre-training for speech with autoregressive predictive coding,” in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3497–3501.
- [23] Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen, “Speech enhancement using long short-term memory based recurrent neural networks for noise robust speaker verification,” in *2016 IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 305–311.
- [24] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur, “A study on data augmentation of reverberant speech for robust speech recognition,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5220–5224.