# SELF-SUPERVISED SPEECH REPRESENTATION LEARNING FOR KEYWORD-SPOTTING WITH LIGHT-WEIGHT TRANSFORMERS

*Chenyang Gao*[⋆] *Yue Gu*[†] *Francesco Caliva*[†] *Yuzong Liu*[†]

[⋆]Rutgers, The State University of New Jersey
[†]Alexa Perceptual Technologies, Amazon

## ABSTRACT

Self-supervised speech representation learning (S3RL) is revolutionizing the way we leverage the ever-growing availability of data. While S3RL related studies typically use large models, we employ light-weight networks to comply with tight memory of compute-constrained devices. We demonstrate the effectiveness of S3RL on a keyword-spotting (KS) problem by using transformers with 330k parameters and propose a mechanism to enhance utterance-wise distinction, which proves crucial for improving performance on classification tasks. On the Google speech commands v2 dataset, the proposed method applied to the Auto-Regressive Predictive Coding S3RL led to a 1.2% accuracy improvement compared to training from scratch. On an in-house KS dataset with four different keywords, it provided 6% to 23.7% relative false accept improvement at fixed false reject rate. We argue this demonstrates the applicability of S3RL approaches to light-weight models for KS and confirms S3RL is a powerful alternative to traditional supervised learning for resource-constrained applications.

***Index Terms—*** Self-supervised speech representation learning; keyword-spotting; on-device classification; light deep learning

## 1. INTRODUCTION

With the prosperous growth of deep learning, scarcity of annotated data is considered a bottleneck in model training. The general consensus is that abundant amounts of data favor model generalization. However, collecting large amounts of labeled data is time consuming and error-prone. Hence, the interest in developing methods to effectively leverage unannotated data has increased. Self-supervised learning (SSL) is a widely adopted solution in Natural Language Processing (NLP) [1, 2] and Computer Vision (CV) [3, 4]. In speech domain, non-annotated data are commonly used to pre-train models based on self-supervised speech representation learning (S3RL) [5]. S3RL methods could be categorized into generative loss-based [6, 7, 8], latent unit prediction-based [9, 10], discriminative learning-based approaches [11, 12, 13] or combinations thereof [14]. These methods are capable of learning meaningful representation of speech. They can use the gained knowledge for instance to infer missing features given a context or to classify data even when relevant features of the input have been masked out. Learned representations from S3RL have been shown to generalize to keyword-spotting (KS) [5, 15] but limitations still exist. S3RL methods require training models with millions of parameters to be able to generate good quality speech representation and this requirement cannot be accommodated for models deployed on mobile and edge devices. Furthermore, S3RL methods have mostly been designed for automatic speech recognition (ASR) applications and have been neglecting the need for learning different representations among utterances (or utterance-wise distinction), which is critical for classification tasks such as KS, sentiment analysis and speaker identification. To overcome the aforementioned limitations, we employed S3RL on a 330K parameters transformer for KS on-edge. We studied Auto-Regressive Predictive Coding (APC), Masked Predictive Coding (MPC) and Contrastive Learning (CL). A concurrent work used SSL for KS [16]; however, their approach is based on Data2Vec [17], which is a general SSL method. Instead, we focused on methods specific to learning speech related representations and used a transformer with half the number of parameters of the smallest model used in [16]. After pre-training, we added a linear classifier to the light-weight transformer and fine-tuned it to a KS down-stream task. Furthermore, we introduced a novel two-step contrastive-learning mechanism to facilitate learning more diverse utterance-level representations. Our proposed approach can be easily plugged into any S3RL method. We evaluated performance on the publicly available Google speech commands v2 dataset [18] as well as in an in-house KS dataset. Results showed that when applied to APC S3RL, the proposed approach achieved 1.2% accuracy improvement compared to training from scratch on Google Commands V2 35 classes classification. In an in-house KS dataset comprised of four keywords, our method provided 6% to 23.7% relative false accept reduction at fixed false reject rate. In summary, our contributions can be summarized as follows:

- We explored S3RL with light-weight transformers and demonstrated its effectiveness for KS;
- We devised a method which can be easily applied to any S3RL method to improve utterance-wise distinction. This led to a more accurate KS system;
- To the best of our knowledge, we are the first to conduct an extensive and fair comparison among S3RL methods on a common benchmark, and draw conclusions on which S3RL approach is best suited for pre-training light-weight models.

## 2. BACKGROUND

S3RL has proven effective in pre-training large models for speech-related tasks. We focus on three of these methods. **Auto-regressive Predictive Coding** [6] is a generative-loss based SR3L approach. It allows training auto-regressive models, meaning that by using information from past timestamps they can predict future information. APC relies on the training objective function reported in eq. 1, where $\mathbf{X}$=($x_1$, $x_2$,...,$x_T$) is a target sequence, $\mathbf{Y}$=($y_1$, $y_2$,...,$y_T$) is a predicted sequence and $T$ represents the total sequence length. APC

Chenyang Gao performed this work while interning at Amazon.

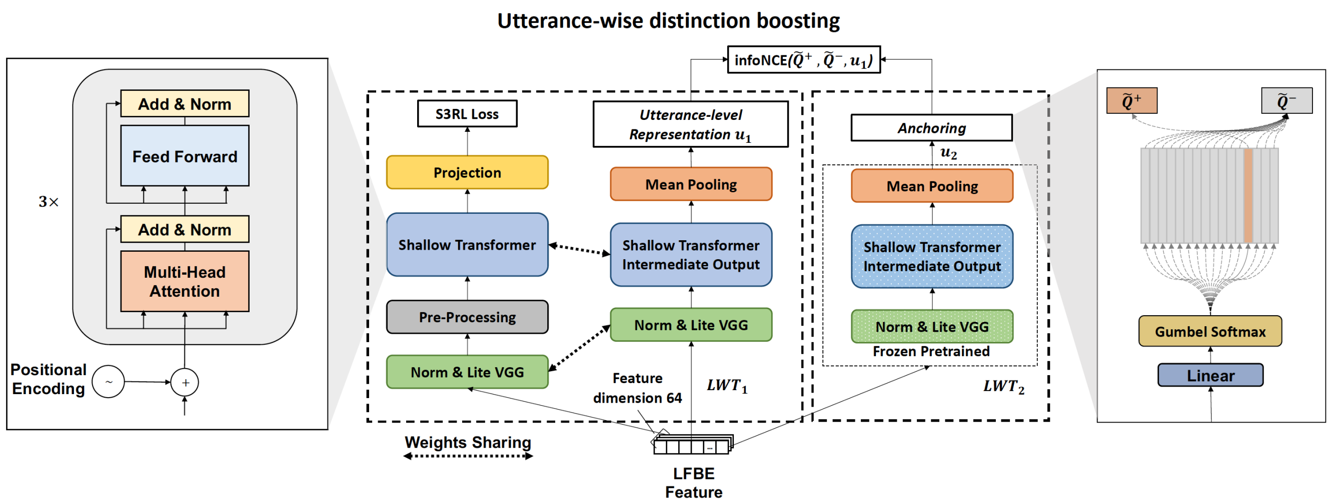


**Fig. 1**. Proposed two steps utterance-wise contrastive-learning. The S3RL block performs S3RL on a task. The utterance-wise distinction boosting block helps locating the anchor class for utterance level representation using utterance-wise boosting, which leverages the pre-trained S3RL model. Please refer to section 3.2 for further details.

predicts frames $n$ steps ahead of a current frame.

$$\mathscr{L}_{APC} = \sum_{i=1}^{T-n} \|x_{i+n} - y_i\| \tag{1}$$

**Masked Predictive Coding** [7, 8] is another generative-loss based SR3L approach. It uses bi-directional context to reconstruct masked features. MPC is generally trained by minimizing a loss as in eq. 2

$$\mathscr{L}_{MPC} = \sum_{i=1}^{T} w_i \|x_i - y_i\| \tag{2}$$

where $w$ weighs the contribution of masked and unmasked regions.
**Contrastive-learning** [11, 13] is a discriminative learning-based S3RL method. It maximizes the distance between the embeddings of positive and negative sample-pairs. Same as [13], we used vector quantization [19] to determine the centroids of different anchor classes. The training objective function was a weighted sum ($\beta$=0.1) between the info Noise-Contrastive Estimation (infoNCE) [11] and a diversity loss ($\mathscr{L}_D$) [13].

$$\mathscr{L}_{CL} = -\sum_{i=1}^{T} w_i \log \frac{\exp(\Phi(y_i, q_i)/\kappa)}{\sum_{\tilde{q}\in Q_p} \exp(\Phi(y_i, \tilde{q})/\kappa)} + \beta\mathscr{L}_D \tag{3}$$

In eq. 3, $Q_p$ refers to acoustic-unit level centroids quantized by using a Gumbel softmax operation [20]; $q_i$ refers to the positive sample of $y_i$, calculated using the $i$-th position in the original input. We consider the remaining codebook entries as negative samples and a cosine similarity distance function $\Phi(\cdot)$. As in MPC, $w$ is a function that weighs the contribution of masked and unmasked regions. $\kappa$ is a temperature scaling factor set to 0.1 as in [13]. $\mathscr{L}_D$ encourages the usage of different code entries and is defined as $\mathscr{L}_D = 1/GV \sum_{g=1}^{G} \sum_{v=1}^{V} p_{g,v} \log p_{g,v}$, where $G$ is the codebook number, $V$ the entry size of each codebook, $p_{g,v}$ the probability of selecting $v$-th entry of the $g$-th codebook and is calculated using a softmax function across a batch of training sequences.

## 3. PROPOSED APPROACH

Here we introduce the small-footprint transformer model which was used in our experiments. Following that, we present our proposed method to enhance utterance-wise distinctions, which improves a model's ability to differentiate among utterances during S3RL.

### 3.1. Light-weight transformer

Since its introduction in [21], transformer and variants have found vast application in CV and NLP. The majority of these models exceed millions of parameters [1, 13, 8, 11, 7], making them not suitable for KS on compute-constrained devices. Attention-matrix computation on transformers has time complexity $T^2$, where $T$ refers to the input sequence length. Reducing the input size can mitigate such computational demand. We accomplished this by processing the input with a VGG-like model [22] and by using a strided-convolution-based $2\times$ down-sampling factor. To make our transformer mobile-friendly, we limited it to 330K parameters by decreasing both depth and width, making it comparable in size with KS models in [23]. We implemented a number of simple, yet effective modifications to the adopted S3RL methods. In APC, models exploit bidirectional information hence violate the concept of auto-regressive methods where only past information should be used; we introduced an attention mask to ensure our model focused only on past timestamps. In MPC, binary masks are randomly generated to prevent a model from exploiting local smoothness due to continuous properties of acoustic features [7, 8]; we took inspiration from [13] and used learnable mask-embeddings instead. We set weights to 0 and 1.0 for unmasked and masked region respectively to focus on reconstructing missing features.

### 3.2. Improving utterance-wise distinction

[24] used CL to enhance model's ability to learn different characteristics amongst speakers. This speaker-wise CL provided a performance boost on down-stream tasks including speaker identification and keyword-spotting. While [24] focused on speaker-difference, we are interested in learning general and abstract differences among utterances. Similar to past research [5, 15, 24], we captured utterance level differences by processing the S3RL-learned features with a mean-pooling operator followed by a linear transformation. In addition, we propose a two-step mechanism to enhance utterance-wise distinction. Our approach is schematically depicted in Fig. 1. The method comprises two blocks: the first block is a general S3RL block, which includes a light-weight transformer ($LWT_1$). In principle, $LWT_1$ could be trained using any S3RL methods. Inputs

**Table 1**. Experimental results obtained on Google Speech Command V2 and on an in-house keyword-spotting datasets.

| | Training strategy | | Google speech commands v2 dataset | | In-house dataset | |
|---|---|---|---|---|---|---|
| **Approach** | **PT** | **FT** | **PT/FT** | **Accuracy** | **PT/FT** | **Relative FAR** |
| Baseline_MLP | × | Decoder only | -/GC_V2 | 39.2% | -/- | - |
| Baseline_LT | × | Encoder+Decoder | -/GC_V2 | 94.6% | -/IH_1.6K | 1.0 |
| LT_APC$_{mlp}$ | APC | Decoder only | LS_960/GC_V2 | 61.5% | IH_15K/IH_1.6K | 1.371 |
| LT_APC | APC | Encoder+Decoder | LS_960/GC_V2 | 95.3% | IH_15K/IH_1.6K | 0.827 |
| LT_MPC | MPC | Encoder+Decoder | LS_960/GC_V2 | 95.0% | IH_15K/IH_1.6K | 0.838 |
| LT_CL | CL | Encoder+Decoder | LS_960/GC_V2 | 94.9% | IH_15K/IH_1.6K | 0.854 |
| **LT_APC+** | APC_uwdb | Encoder+Decoder | LS_960/GC_V2 | **95.8%** | IH_15K/IH_1.6K | **0.818** |
| LT_MPC+ | MPC_uwdb | Encoder+Decoder | LS_960/GC_V2 | 95.3% | IH_15K/IH_1.6K | 0.831 |
| LT_CL+ | CL_uwdb | Encoder+Decoder | LS_960/GC_V2 | 95.3% | IH_15K/IH_1.6K | 0.847 |

Legend **GC_V2**=Google Speech Command V2 dataset; **LS_960**=Librispeech 960hrs dataset; **IH_15K**=in-house 15K hours of KS audio used as unannotated data; **IH_1.6K**=in-house 1.6K hours of annotated KW$_1$ audio sub-dataset. **uwdb**=utterance-wise distinction boosting; **Relative FAR**=relative false acceptance rate compared to baseline at fixed false rejection rate; **PT/FT**=pre-training/fine-tuning; **Decoder**=one MLP layer only; **Encoder**= Transformer blocks.

are pre-processed as described in section 2 to accommodate specific S3RL input requirements. The second block includes another transformer model ($LWT_2$). $LWT_2$ is pre-trained with the same S3RL method as $LWT_1$. Weights of $LWT_2$ are frozen, so that $LWT_2$ is used as a feature extractor. Inputs to $LWT_2$ are not pre-processed to obtain the utterance-level representation. The training procedure is as follows: *i)* intermediate output features are extracted at the second layer of $LWT_1$ and $LWT_2$, *ii)* these features are processed by a mean-pooling operator to generate utterance-level representation $u_1$ for $LWT_1$ and $u_2$ for $LWT_2$ and *iii)* $u_2$ representations are fed into an anchoring block to generate positive ($\tilde{Q}^{+}$) and negative ($\tilde{Q}^{-}$) sample-pairs based on vector quantization. With regard to $LWT_1$, weights are tuned in a multi-task learning fashion, simultaneously solving S3RL and utterance-wise contrastive learning. For utterance-wise CL, we minimize a combination of the infoNCE contrastive learning loss function ($\mathscr{L}_{utt}$) and the diversity loss $\mathscr{L}_D$ with $\beta$=0.1:

$$\mathscr{L}_{utt} = -\log \frac{\exp\left(\Phi(u_1, \tilde{Q}^{+})/\kappa\right)}{\sum_{\tilde{q}\in Q} \exp\left(\Phi(u_1, \tilde{q})/\kappa\right)} + \beta \mathscr{L}_D \tag{4}$$

The complete training objective function used in our experiments is reported in eq. 5 and to maintain self-supervised learning properties, we set $\alpha$ to 0.9.

$$\mathscr{L} = \alpha \mathscr{L}_{S3RL} + (1-\alpha)\mathscr{L}_{utt} \tag{5}$$

## 4. EXPERIMENTAL STUDY

### 4.1. Datasets

We used three datasets throughout our experimentation: one is comprised of de-identified audio files from our production data and the others are open-source datasets, widely used among the speech-processing research community. All data were processed in the form of 64-D LFBE spectrogram. These were formulated by using an analysis window and shift-size of 25ms and 10ms respectively.

**Librispeech dataset** This dataset contains speech in English, sampled at 16kHz. We used training subsets of Librispeech [25] for self-supervised pre-training, creating 960 hours of training corpus.

**Google speech commands v2 dataset** This dataset is comprised of 105,829 one second-long utterances of 35 keywords [18] and was used for KS down-streaming task.

**In-house keyword-spotting dataset** We collected 20K hours worth of de-identified audio containing four keywords (16.6K hours with KW$_1$, 1.6K hours with KW$_2$, 833 hours with KW$_3$ and 1K hours with KW$_4$), recorded at different front-end conditions and used them for training purpose. We used KW$_1$ for single keyword experiments, where we split the datasets into 15K/1.6K hours for pre-training and fine-tuning, respectively. In multi-wakeword experiments, we used the entire dataset. Our collected test data consisted of 8.5K hours with KW$_1$, 1K hours with KW$_2$, 444 hours with KW$_3$ and 500 hours with KW$_4$ audio streams. The keywords were semi-automatically annotated and inspected for quality check.

### 4.2. Training setup

**S3RL configuration** For APC, we set the number of predicted future frames $n$=8. This is equivalent to $n$=5 in [6], since the light VGG-style model which we used has a receptive-field of 7 frames. With respect to the MPC masking operation, we masked out 50% of the frames and replaced them with mask-embedding, unlike [7, 8]. With regard to CL, we adopted a single codebook with 64 entries to accommodate our limited model capacity. Similarly, for utterance-wise CL, we used a single codebook with 32 entries.

**Training configuration** S3RL training objectives were defined in section 2. As shown in Fig 1, we applied three transformer blocks as the encoder and one MLP layer (projection) as the decoder. To fine-tune our models on the KS task, we minimized the cross-entropy loss function. We compared the performance with freezing and unfreezing setups as in [5] and [1] respectively. Due to the model's limited capacity, we used the output from final layers as representations instead of a weighted sum of output from different layers [5, 26, 24]. We did not apply SpecAug [27] as this could obscure the the effectiveness of S3RL and add additional variables to account for. We ran pre-training for 20 epochs and fine-tuned our models for 10 additional epochs on the KS task, using an Adam optimizer [28] with initial learning rate set to 1e-3 and per-epoch exponential weight decay of rate 0.95. Each epoch consisted of 5000 steps with batch size equal to 2000, distributed across 8 NVIDIA Tesla V100 GPUs.

## 5. RESULTS AND DISCUSSION

### 5.1. Google speech commands v2 dataset

**S3RL is feasible in light-weight transformers** In table 1, we compare model performance at various settings, in terms of accuracy, as common practice in the literature. We found that self-supervised pre-training with model fine-tuning achieved better performance than training from scratch (Baseline_LT), irrespective of the S3RL

**Table 2**. KS results of the ablation study conducted on an in-house dataset. Performance is reported in terms of relative false acceptance rate (FAR) to the baseline model at fixed false rejection rate (FRR).

| | Fine-tuning dataset (hours) | | | |
|---|---|---|---|---|
| Approach | 166 | 416 | 833 | 1.6K |
| Baseline_LT | 1.35 | 1.27 | 1.09 | 1.0 |
| LT_MPC+ | 1.20 | 1.12 | **0.901** | 0.831 |
| LT_CL+ | 1.31 | 1.12 | **0.903** | 0.847 |
| **LT_APC+** | 1.19 | **1.07** | **0.895** | 0.818 |
| | $n$ future in APC | | | |
| Approach | 5 | 8 | 10 | 20 |
| LT_APC+ | 0.910 | **0.818** | 0.852 | 0.973 |
| | Mask proportion in MPC | | | |
| Approach | 15% | 25% | 50% | 75% |
| LT_MPC+ | 0.953 | 0.878 | **0.831** | 0.950 |
| | Codebook size in CL | | | |
| Approach | 32 | 64 | 128 | 256 |
| LT_CL+ | 0.872 | **0.847** | 0.859 | 0.862 |

method. This demonstrates S3RL is feasible on KS with tiny footprint models, using large amounts of unlabeled data. Furthermore, we observed LT_APC$_{mlp}$ outperformed the baseline multi-layer perceptron (MLP) model (Baseline_MLP). This demonstrates S3RL facilitates training light-weight models. Freezing the entire encoder and fine-tuning the decoder's weights (LT_APC$_{mlp}$) degraded performance, compared to training from scratch (Baseline_LT). We suppose this is because the model which we used for pre-training was not large enough to be able to learn general speech representations. The performance improvement of LT_APC compared to Baseline_LT indicates that down-stream tasks can benefit from pre-training, likely because it provides a better network initialization. From table 1, the LT_APC provided the largest performance improvement (0.7% accuracy) among the remaining S3RL methods.

**Improving utterance-wise distinction positively impacts KS** We conducted experiments using our proposed method to improve utterance-wise representations. We used pre-trained APC, MPC and CL models and denote these configurations with "+". As reported in table 1, APC+, MPC+ and CL+ outperformed original light-weight S3RL methods respectively. Specifically, LT_APC+ provided 0.6% accuracy improvement compared to LT_APC and 1.2% accuracy improvement compared to Baseline_LT. We suggest this demonstrates the benefits of the proposed second-step utterance-wise distinction boosting. Intuitively, if we used larger utterance-wise distinction models for S3RL boosting, utterance level representations would improve even further. We leave this investigation for future work.

### 5.2. In-house keyword-spotting dataset

We conducted performance analysis on our in-house dataset by means of false acceptance rate (FAR) relative to the baseline model at fixed false rejection rate (FRR). For a specific keyword, the FRR is the ratio between false negative and true positive samples at the operating point (OP) of the baseline model. We identified an OP at which the proposed approach showed similar FRR and used that OP to calculate the corresponding FAR, which is the ratio between false positives and true negatives. As reported in table 1, APC and APC+ achieved better performance among S3RL methods. The LT_APC showed 17.3% relative FAR improvement (from 1.0 to 0.827) compared to Baseline_LT. The LT_APC+ further reduced the relative FAR to 0.818 from 0.827. These results demonstrate

**Table 3**. Multi-keyword-spotting results on an in-house dataset. Performance is reported in terms of relative false acceptance rate (FAR) to the baseline model at fixed false rejection rate (FRR).

| Approach | $KW_1$ | $KW_2$ | $KW_3$ | $KW_4$ |
|---|---|---|---|---|
| Training dataset (hours) | 16.6K | 1.6K | 833 | 1.1K |
| Baseline_LT | 1.0 | 1.0 | 1.0 | 1.0 |
| LT_MPC+ | **0.906** | 0.915 | 0.878 | 0.960 |
| LT_CL+ | 0.919 | 0.920 | 0.886 | 0.957 |
| **LT_APC+** | 0.910 | **0.876** | **0.763** | **0.940** |

that S3RL can be effective even when using light-weight models and large scale datasets. On the other hand, these results show that CL-based S3RL did not perform on-par with the other S3RL methods. We speculate this is due to the limited model capacity, which constrained the model learning capability. We further conducted an ablation study on the same in-house $KW_1$ dataset and report results in table 2. When we fine-tuned with just 50% of the data, we performed on-par or even outperformed the model trained from scratch (Baseline_LT). In our ablation study, using 1.6K hours worth of $KW_1$ training audio for fine-tuning, APC, MPC and CL based approaches achieved best performance with future shift $n$=8, 50% mask proportion and 64 entries in the codebook. We extended the experiment to multiple keywords from our in-house dataset. Instead of using a single keyword during pre-training, we combined the first three keyword data for pre-training and left the fourth keyword out as an independent dataset to prove the out of domain effectiveness in S3RL. We fine-tuned the model for each keyword independently using the corresponding audio samples. Results in table 3 show that the LT_APC+ performed better than the Baseline_LT on all four keywords task. It achieved 12,4% and 23.7% relative FAR improvement on $KW_2$ and $KW_3$. Even without in-domain data in the pre-training dataset, the model still achieved a 6% relative FAR gain on $KW_4$. These results indicate that when data are scarce, down-stream tasks can benefit from S3RL.

## 6. CONCLUSION AND FUTURE WORK

In this paper, we explored the feasibility of using self-supervised representation learning on small-footprint models. Specifically, we pre-trained light-transformers by means of three S3RL methods, such as APC, MPC and CL and subsequently fine-tuned the transformers on KS down-stream tasks. In our experimental study, all three S3RL methods showed better performance compared to training from scratch. Pre-training using APC provided the best results on both public and in-house datasets. In addition, we proposed a method that combines S3RL and CL to boost representation learning by extracting more diverse utterance-level representations. We showed that the proposed combination led to better performance on the down-stream KS task. It is important to remark that our method can be easily implemented and added to any S3RL methods. As part of the future work, additional speech representation learning methods should be investigated, as well as alternative large models to extract utterance-level representation. It would be interesting to explore the effect of learning representations by means of a model as proposed by [29], as opposed to the mean pooling approach which was adopted in this work.

## 7. REFERENCES


[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *NAACL*, 2019.

[2] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, “Deep contextualized word representations,” in *ACL*, 2018.

[3] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick, “Momentum contrast for unsupervised visual representation learning,” in *CVPR*, 2020.

[4] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick, “Masked autoencoders are scalable vision learners,” in *CVPR*, 2022.

[5] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al., “Superb: Speech processing universal performance benchmark,” in *arXiv:2105.01051*, 2021.

[6] Yu-An Chung and James Glass, “Generative pre-training for speech with autoregressive predictive coding,” in *ICASSP*, 2020.

[7] Dongwei Jiang, Xiaoning Lei, Wubo Li, Ne Luo, Yuxuan Hu, Wei Zou, and Xiangang Li, “Improving transformer-based speech recognition using unsupervised pre-training,” in *arXiv:1910.09932*, 2019.

[8] Andy T Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee, “Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders,” in *ICASSP*, 2020.

[9] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[10] Heng-Jui Chang, Shu-wen Yang, and Hung-yi Lee, “Distilhubert: Speech representation learning by layer-wise distillation of hidden-unit bert,” in *ICASSP*, 2022.

[11] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, “Representation learning with contrastive predictive coding,” in *NeurIPS*, 2018.

[12] Shaoshi Ling, Yuzong Liu, Julian Salazar, and Katrin Kirchhoff, “Deep contextualized acoustic representations for semi-supervised speech recognition,” in *ICASSP*, 2020.

[13] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” in *NeurIPS*, 2020.

[14] Shaoshi Ling and Yuzong Liu, “Decoar 2.0: Deep contextualized acoustic representations with vector quantization,” in *arXiv:2012.06659*, 2020.

[15] Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass, “Ssast: Self-supervised audio spectrogram transformer,” in *AAAI*, 2022.

[16] Holger Severin Bovbjerg and Zheng-Hua Tan, “Improving label-deficient keyword spotting using self-supervised pre-training,” in *arXiv:2210.01703*, 2022.

[17] Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli, “Data2vec: A general framework for self-supervised learning in speech, vision and language,” in *arXiv:2202.03555*, 2022.

[18] Pete Warden, “Speech commands: A dataset for limited-vocabulary speech recognition,” in *arXiv:1804.03209*, 2018.

[19] Herve Jegou, Matthijs Douze, and Cordelia Schmid, “Product quantization for nearest neighbor search,” in *IEEE PAMI*, 2010.

[20] Eric Jang, Shixiang Gu, and Ben Poole, “Categorical reparameterization with gumbel-softmax,” in *ICLR*, 2017.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in *NeurIPS*, 2017.

[22] Karen Simonyan and Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *arXiv:1409.1556*, 2014.

[23] Mohammad Omar Khursheed, Christin Jose, Rajath Kumar, Gengshen Fu, Brian Kulis, and Santosh Kumar Cheekatmalla, “Tiny-crnn: Streaming wakeword detection in a low footprint setting,” in *ASRU*, 2021.

[24] Sanyuan Chen, Yu Wu, Chengyi Wang, Zhengyang Chen, Zhuo Chen, Shujie Liu, Jian Wu, Yao Qian, Furu Wei, Jinyu Li, et al., “Unispeech-sat: Universal speech representation learning with speaker aware pre-training,” in *ICASSP*, 2022.

[25] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *ICASSP*, 2015.

[26] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al., “Wavlm: Large-scale self-supervised pre-training for full stack speech processing,” in *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[27] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, “Specaugment: A simple data augmentation method for automatic speech recognition,” in *arXiv:1904.08779*, 2019.

[28] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” in *arXiv:1412.6980*, 2014.

[29] Themos Stafylakis, Ladislav Mosner, Sofoklis Kakouros, Oldrich Plchot, Lukas Burget, and Jan Cernocky, “Extracting speaker and emotion information from self-supervised speech models via channel-wise correlations,” in *arXiv:2210.09513*, 2022.