Self-Supervised Speech Representation Learning: A Review

Abdelrahman Mohamed, *Member, IEEE*, Hung-yi Lee, Lasse Borgholt, Jakob D. Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, *Senior Member, IEEE*, Lars Maaløe, Tara N. Sainath, and Shinji Watanabe, *Senior Member, IEEE*

***Abstract*—Although supervised deep learning has revolutionized speech and audio processing, it has necessitated the building of specialist models for individual tasks and application scenarios. It is likewise difficult to apply this to dialects and languages for which only limited labeled data is available. Self-supervised representation learning methods promise a single universal model that would benefit a wide variety of tasks and domains. Such methods have shown success in natural language processing and computer vision domains, achieving new levels of performance while reducing the number of labels required for many downstream scenarios. Speech representation learning is experiencing similar progress in three main categories: generative, contrastive, and predictive methods. Other approaches rely on multi-modal data for pre-training, mixing text or visual data streams with speech. Although self-supervised speech representation is still a nascent research area, it is closely related to acoustic word embedding and learning with zero lexical resources, both of which have seen active research for many years. This review presents approaches for self-supervised speech representation learning and their connection to other research areas. Since many current methods focus solely on automatic speech recognition as a downstream task, we review recent efforts on benchmarking learned representations to extend the application beyond speech recognition.**

***Index Terms*—Self-supervised learning, speech representations.**

Manuscript received 12 April 2022; revised 4 August 2022; accepted 24 August 2022. Date of publication 15 September 2022; date of current version 14 October 2022. The work of Jakob D. Havtorn was supported in part by Innovation Fund Denmark through Industrial Ph.D. Programme under Grant 0153-00167B. The work of Lasse Borgholt was supported in part by Innovation Fund Denmark thorugh Industrial Ph.D. Programme under Grant 8053-00184B. The guest editor coordinating the review of this manuscript and approving it for publication was Prof. Xiao-Ping Zhang. *(Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, and Jakob D. Havtorn contributed equally to this work.) (Corresponding author: Hung-yi Lee.)*

Abdelrahman Mohamed and Shang-Wen Li are with the Meta, Menlo Park, CA 94025 USA (e-mail: abdo@fb.com; shangwel@fb.com).

Hung-yi Lee is with the Department of Electrical Engineering and Department of Computer Science Information Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: hungyilee@ntu.edu.tw).

Lasse Borgholt is with the Corti AI and Department of Computer Science, University of Copenhagen, 1165 København, Denmark (e-mail: lb@corti.ai).

Jakob D. Havtorn and Lars Maaløe are with the Corti AI and Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark (e-mail: jdha@dtu.dk; lm@corti.ai).

Joakim Edin is with the Corti AI, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark (e-mail: je@corti.ai).

Christian Igel is with the Department of Computer Science, University of Copenhagen, 1165 København, Denmark (e-mail: igel@di.ku.dk).

Katrin Kirchhoff is with the AWS AI Labs, Amazon, Seattle 98121 USA (e-mail: katrinki@amazon.com).

Karen Livescu is with the Toyota Technological Institute at Chicago, Chicago, IL 60615 USA (e-mail: klivescu@ttic.edu).

Tara N. Sainath is with the Google, Inc., New York, NY 1011 USA (e-mail: tsainath@google.com).

Shinji Watanabe is with the Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: shinjiw@ieee.org).

Digital Object Identifier 10.1109/JSTSP.2022.3207050

I. Introduction

Over the past decade, deep learning approaches have revolutionized speech processing through a giant leap in performance, enabling various real-world applications. Supervised learning of deep neural networks has been the cornerstone of this transformation, offering impressive gains for scenarios rich in labeled data [1], [2], [3]. Paradoxically, this heavy reliance on supervised learning has restricted progress in languages and domains that do not attract the same level of labeling investment.

To overcome the need for labeled data, researchers have explored approaches that use unpaired audio-only data to open up new industrial speech use-cases and low-resource languages [4], [5], [6]. Inspired by how children learn their first language through listening and interacting with family and surroundings, scientists seek to use raw waveforms and spectral signals to learn speech representations that capture low-level acoustic events, lexical knowledge, all the way to syntactic and semantic information. These learned representations are then used for target downstream applications requiring a minimal number of labeled data [7], [8], [9]. Formally, representation learning refers to algorithms for extracting latent features that capture the underlying explanatory factors for the observed input [9].

Representation learning approaches are generally considered examples of *unsupervised learning*, which refers to the family of machine learning methods that discover naturally occurring patterns in training samples for which there are no pre-assigned labels or scores [10]. The term "unsupervised" is used to distinguish this family of methods from "supervised" approaches, which assign a label to each training sample, and "semi-supervised" approaches, which utilize a small number of training samples with labels to guide learning using a larger volume of unlabeled samples. Examples of unsupervised learning techniques include k-means clustering [11], mixture models [12], autoencoders [13], and non-negative matrix factorization [14]. *Self-supervised learning* (SSL) is a fast-growing subcategory of unsupervised learning approaches, which are techniques that utilize information extracted from the input data itself as the

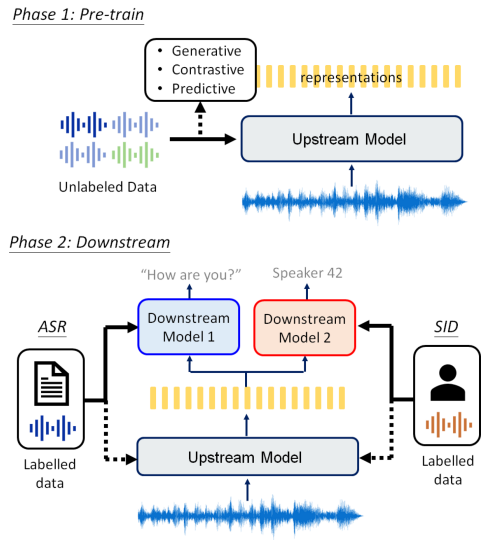

Fig. 1. Framework for using self-supervised representation learning in downstream applications.

label to learn representations useful for downstream tasks. For example, unsupervised k-means clustering doesn't adhere to this definition of self-supervision since it iteratively minimizes the within-cluster variance during learning. In this review, we focus on self-supervised learning approaches.

Fig. 1 outlines self-supervised representation learning in relation to downstream applications. There are two stages in this framework. In the first stage, we use SSL to pre-train a *representation model*, also called an *upstream model* or a *foundation model*. In the second stage, downstream tasks use either the learned representation from the frozen model, or fine-tune the entire pre-trained model in a supervised phase [15]. Automatic speech recognition (ASR) and speaker identification (SID) are examples of downstream applications in Fig. 1.

It is considered desirable for learned speech representations to be disentangled, invariant, and hierarchical. Since spoken utterances contain much richer information than the corresponding text transcriptions—e.g., speaker identity, style, emotion, surrounding noise, and communication channel noise—it is important to learn representations that disentangle these factors of variation. Furthermore, invariance of the learned features to changes in background noise and in the communication channel ensures stability with respect to downstream application scenarios. Learning feature hierarchies at the acoustic, lexical, and semantic levels supports applications with different requirements. For instance, whereas a speaker identification task benefits from a low-level acoustic representation, a speech translation task requires a more semantic representation of the input utterance.

Due to the popularity of SSL, reviews have been published about the technology in general [16], [17], [18] as well as its application to natural language processing (NLP) [19], [20], [21], [22] and computer vision (CV) [23]. Recently, a brief overview with a general focus on speech representation learning was published [24]. However, none of these overviews focus exclusively on SSL for speech processing. Since the speech signal differs greatly from image and text inputs, many theories and technologies have been developed to address the unique challenges of speech. One review addresses speech representation learning based on deep learning models [25], but does not address recent developments in self-supervised learning. This motivates this overview of speech SSL.

The structure of this paper is arranged as follows. Section II briefly reviews the history of speech representation learning, and Section III reviews current speech SSL models. Section IV surveys SSL datasets and benchmarks, and discusses and compares results from different works. Section V analyzes successful SSL approaches and offers insights into the importance of technological innovations. Section VI-B reviews zero-resource downstream tasks that utilize SSL. Finally, Section VII summarizes the paper and suggests future research directions.

II. Historical Context of Representation Learning

In this section we present the historical background of the current surge in self-supervised representation learning methods in the context of two previous waves of research work in the 1990 s and 2000 s. The discussed approaches go beyond speech to describe the overall landscape of machine learning development during the past few decades.

A. Clustering and Mixture Models

Initial research in learning latent speech and audio representations involved simple models in which the training data likelihood was optimized directly or via the expectation–maximization (EM) algorithm.

Early work used simple clustering methods. For example, in work such as [26], [27], word patterns were clustered semi-automatically using techniques such as k-means, after which isolated words were recognized by finding the training cluster closest to the test data.

Through time, modeling techniques improved such that subword units were represented by Gaussian mixture models (GMMs) [28], which facilitated the modeling of more variability in the input data. GMMs were first built for context-independent phonemes; state-clustering algorithms [29] then resulted in GMMs for context-dependent phonemes. Each latent component of these mixture models acted as a template of a prototypical speech frame, making it difficult to handle large volumes of data with diverse characteristics. Furthermore, dynamical models like hidden Markov models (HMMs) [30] allowed for the processing of continuous speech rather than just isolated word recognition. These generative GMM and HMM models were trained by maximizing the likelihood of data given the model, which could be accomplished in either an unsupervised or a supervised manner.

Another line of research focused on extracting speech features from generative models. The main objective here was to render the knowledge learned by generative models accessible to discriminative downstream classifiers, or to map variable-length sequences to fixed-length representations. Feature vectors were derived from the parameters of trained GMM models. In the case of *Fisher vectors*, the features were the normalized gradients of the log-likelihood with respect to the model parameters (mixture weights, means, and variances) of the Gaussian mixtures. An extension of this approach (likelihood ratio score space) used

the derivative of the log-likelihood ratio of two models, e.g., a background model and a foreground model. Examples of their use in speech processing include speech recognition [31], [32] and speaker recognition [33]. Subsequent techniques in speaker and language verification [34], [35] similarly extracted parameters (concatenated means) from trained background GMMs as representations that were then combined with low-rank projections of speaker/session- or language-specific vectors.

B. Stacked Neural Models

More recently, representation learning has seen a shift of focus towards neural models, which, compared to GMMs and HMMs, offer distributed representations with more capacity to model diverse input signals into efficient latent binary codes. Examples of early techniques include restricted Boltzmann machines (RBM) [15], denoising autoencoders [36], noise contrastive estimation (NCE) [37], sparse coding [38], [39], [40], and energy-based methods [41]. Many of these techniques have also been applied to CV and NLP problems, which provided inspiration for their application to speech.

Higher-capacity neural models were achieved by stacking several neural network layers to build progressively higher-level concept representations. However, these deeper networks also increased the training complexities. For example, approximate training methods such as contrastive divergence [42] were a practical technique to streamline RBM training. Furthermore, deep networks had non-convex objective functions, which often resulted in long training times compared to GMMs, which are trained using full batches instead of mini-batch learning.

C. Learning Through Pretext Task Optimization

A more recent trend is learning networks that map the input to desired representations by solving a *pretext* task. Such studies have several characteristics: (1) All layers are trained end-to-end to optimize a single pretext task instead of relying on layer-wise pre-training (2) Past stacked networks typically had only a few layers, but very deep networks with more than ten layers are now common. (3) It is common to evaluate a representation model on a wide range of tasks. For example, in NLP, a representation model is usually assessed on GLUE, which comprises nine tasks [43], whereas in speech, a representation model can be evaluated on SUPERB, which comprises ten tasks [44], as described in detail in Section IV-E.

The cornerstone of this third wave is the design of a pretext task, which allows the model to efficiently leverage knowledge from unlabeled data. The pretext task should be challenging enough for the model to learn high-level abstract representations and not be so easy as to encourage the exploitation of low-level shortcuts. Early breakthroughs included end-to-end learning of deep neural architectures via pretext tasks for restoring the true color of black-and-white images [45], joint learning of latent representations and their cluster assignments [46], and the prediction of the relative positions of image patches [47]. Other popular approaches include variational autoencoders (VAEs) [48], [49]. While typical autoencoders learn data representations using unsupervised objectives by reconstructing the input after passing it through an information bottleneck, VAEs estimate a neural model of a probability density function (pdf) that approximates the unknown "true" distribution of the observed data, for which we only have access to independently identically distributed (iid) samples. It is also important to mention dynamical VAEs [50], which is an extension of VAE for sequential data such as speech.

In the SSL context, a pretext task related to autoencoding is to generate an object from its partial information. Such tasks are widely used in NLP, for example, using the previous tokens in a sentence to predict the next token such as in ELMo [51], the GPT series [52], and Megatron [53], or predicting the masked tokens in a sentence such as with the bidirectional encoder representations from Transformers (BERT) series [54], [55]. Another common pretext task in the third wave is contrastive learning [56], in which a model learns to identify a target instance from a set of negative samples. This approach has become especially popular in the CV context [57], [58], [59], [60]. In this survey, we will mainly focus on techniques for pretext task optimization for speech processing, and discuss these techniques in detail in Section III.

D. Other Related Work

A closely related area of research that is not covered in this review is semi-supervised pre-training methods such as pseudo-labeling (that is, self-training). Pseudo-labeling (PL) relies on a supervised teacher model to label a large volume of speech-only data, which is then used to augment the initial labeled data to train a student model [4], [5], [6], [61]. PL has been successful and widely adopted in the speech community since the 1990 s. Other proposed variations of PL include augmenting speech-only data with noise to improve robustness, iterating over the PL process to improve teacher labeling quality, and training student models with more parameters than their original teachers to capture the complexities in vastly larger speech-only data [62], [63], [64]. Both SSL and PL leverage unlabeled speech-only data. One distinguishing factor in PL is the utilization of supervised data for a specific task during model pre-training, which limits the model's focus to a single (or at best a few) downstream tasks. SSL, in turn, is an attempt to learn task-agnostic representations to benefit a wide range of tasks.

Transfer learning (TL) is another closely related area of research for pre-training speech models. TL transfers knowledge captured by models trained on one task to different but related tasks [65]. The past few decades have seen active research on TL and its extension to multitask learning for more general representations. Multilingual and cross-lingual supervised models have proven superior in low-resource speech recognition tasks [66]. SSL can be regarded as a type of TL because knowledge learned from pre-training is used for different downstream tasks. This survey paper focuses on SSL, and not all TL technologies for speech. One survey indeed addresses TL for speech processing [67] but does not include current SSL technologies for speech.

III. Speech Representation Learning Paradigms

Due to the characteristics of speech, SSL pretext tasks developed for CV and NLP may not directly apply to speech. Below we summarize the characteristics of speech as compared to CV and NLP.

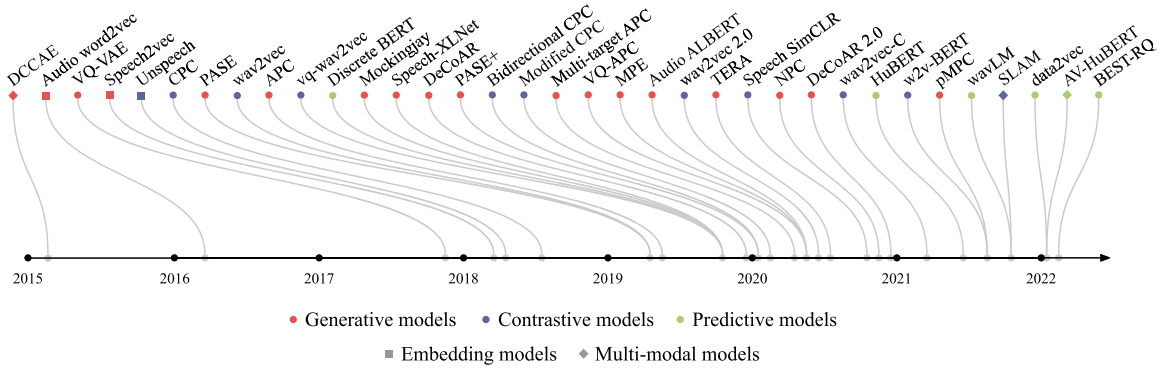

Fig. 2. A selection of models listed according to first publication date on arXiv or conference submission date when this clearly precedes the former. The models are categorized as generative, contrastive, or predictive. In addition, some models are characterized as embedding models or multi-modal models, although most learn frame-level representations from speech only. Some models use a mixture of generative and contrastive tasks. For instance, PASE and PASE+ use a multi-task setup, but find that generative tasks are the most important for downstream task performance [69].

- *Speech is a sequence.* Unlike CV, in which an image usually has a fixed size representation, it is natural to represent a speech utterance as a variable-length sequence. Therefore, pretext tasks developed for CV cannot generally be directly applied to speech.
- *Speech is a long sequence without segment boundaries.* Both text and speech can be represented as sequences. From this viewpoint, it is natural to apply learning approaches developed for text directly to speech. In NLP, morpheme-like tokens are widely used as sequence units in pre-training. The standard BERT takes 512 morpheme-like tokens as input, usually covering a paragraph including several sentences. However, speech signals consist of sound pressure measurements with thousands of samples per second, resulting in sequences much longer than those for text. Even spectral representations which reduce the sequence length can have hundreds of frames per second. Processing such sequences with typical neural network architectures like Transformers can result in problems with running time and memory requirements. One could gather consecutive frames to form shorter segments, but unlike text, there is no obvious segmentation for unlabeled speech.
- *Speech is continuous.* In NLP, it is common to use a pretext task that models a categorical distribution of masked or future inputs. Since text is easily broken down into individual tokens such as words, subwords, or characters, it is straightforward to define a finite vocabulary for such tasks. However, this idea does not apply to speech modeling because speech signals are continuous; in this sense there is no such thing as a speech vocabulary.
- *Speech processing tasks are diverse.* Building generalizable self-supervised representation models for diverse speech processing tasks is challenging. Speech contains rich, hierarchical information, and different speech tasks may require mutually orthogonal information. For example, speech recognition requires a model that extracts content information but ignores speaker information; in contrast, speaker recognition requires a model that extracts speaker information but removes content information. Therefore, it is challenging to define a self-supervised model whose representations are suitable for both speech recognition and speaker recognition. Analogous considerations apply within CV and NLP.

In the sections below, we group modern SSL pretext tasks designed for speech into three main categories: *generative* approaches, *contrastive* approaches and *predictive* approaches. Fig. 2 shows a timeline of the models covered in these sections with each model colored according to our categorization. Table I summarizes model pretext tasks along within the categories.

A. Notation

To efficiently describe the different approaches, we use a simple notation. Models are assumed to consist of functions $f(\cdot)$ and $g(\cdot)$, where $f(\cdot)$ denotes the representation model to be used after pre-training and $g(\cdot)$ is an auxiliary module needed only to support the pretext task. For instance, in a classic autoencoder, $f(\cdot)$ would denote the encoder and $g(\cdot)$ the decoder. For more complex models, these functions might consist of several components indicated by sub-indices $f_1(\cdot) \dots f_N(\cdot)$. As we will see, many self-supervised models use masking, which replaces some parts of the input or a hidden representation by zeros or a learned vector. We use $m(\cdot)$ to denote a function that applies such masking to its input. Similar to $g(\cdot)$, this function is only used during pre-training.

Given an acoustic input $X = \{x_1, x_2, \dots, x_T\}$, $f(\cdot)$ outputs a representation $H = \{h_1, h_2, \dots, h_T\}$. The input X may be either the raw waveform samples or a sequence of spectral feature vectors. Both are viable options in practice. For simplicity, we do not distinguish between the two in our notation.

While $f(\cdot)$ always takes an acoustic input, the input to $g(\cdot)$ can be either the acoustic signal or another learned representation. Most importantly, $g(\cdot)$ produces an output that is used for the pretext task but is not used by $f(\cdot)$ to produce the representation H. Hence, $g(\cdot)$ can be discarded after pre-training. Finally, $f(\cdot)$ commonly downsamples the temporal dimension, but again, this is not crucial to understand the models, so consider only a single temporal scale $t \in \{1, \dots, T\}$ for notational convenience.

We use $Q = \{q_1, q_2, \dots, q_T\}$ to denote representations that are quantized via codebook learning. Alternatively, discrete

TABLE I

A SUMMARY OF THE APPROACHES IN THE THREE CATEGORIES OF SELF-SUPERVISED LEARNING. COLUMN (A) LISTS THE NAMES OF THE MODELS AND RELATED REFERENCES, COLUMN (B) DEFINES THE MODEL INPUT, COLUMN (C) DEFINES ANY CORRUPTION OF THE INPUT OR HIDDEN REPRESENTATION, AND COLUMN (D) DEFINES THE TARGET OF THE PRETEXT TASK; THE PRETEXT TASK ITSELF IS DESCRIBED BY THE OVERALL MODEL CATEGORY AND THE MAIN TEXT. $X = \{x_1, x_2, \ldots, x_T\}$ IS THE INPUT SEQUENCE IN WHICH x_t CAN BE AN ACOUSTIC FEATURE VECTOR (E.G., MFCC, FILTERBANK, OR SPECTROGRAM FEATURES) OR A WAVEFORM SAMPLE. $X_{[t_1:t_2]}$ REPRESENTS $\{x_{t_1}, x_{t_1+1}, \ldots, x_{t_2}\}$. $X_{-[t_1:t_2]}$ REPRESENTS X IN WHICH THE SEGMENT $X_{[t_1:t_2]} = \{x_{t_1}, x_{t_1+1}, \ldots, x_{t_2}\}$ IS MASKED. x_t^i REPRESENTS THE i-TH DIMENSION OF x_t. IF x_t IS A FRAME IN A SPECTROGRAM, THEN THE i-TH DIMENSION CORRESPONDS TO A SPECIFIC FREQUENCY BIN. $X^{-[f,f+j]}$ REFERS TO A SPECTROGRAM X WHICH IS MASKED ALONG THE FREQUENCY AXIS FROM THE f-TH TO $(f+j)$-TH BIN. WE INDICATE RANDOM TEMPORAL PERMUTATION OF A SEQUENCE BY INDEXING IT WITH THE SET $\mathcal{P}_t \triangleq \text{PERMUTE}([0, t])$, WHERE PERMUTE$(\cdot)$ RETURNS A PERMUTATION OF THE GIVEN LIST. WE INDICATE DATA AUGMENTATION (E.G., REVERBERATION) BY THE FUNCTION AUGMENT$(\cdot)$

Model (a)	**Input** (b)	**Corruption** (c)	**Target** (d)
	GENERATIVE MODELS		
Audio Word2vec [118], VQ-VAE [75]	X	-	X
Speech2Vec [114], Audio2Vec [116] - skip-gram	$X_{[t_1,t_2]}$	-	$X_{[t_0,t_1]}$,$X_{[t_2,t_3]}$
Speech2Vec [114], Audio2Vec [116] - cbow	$X_{[t_0,t_1]}$,$X_{[t_2,t_3]}$	-	$X_{[t_1,t_2]}$
PASE [69], PASE+ [113][1]	X	-	Different modalities of X
APC [92], [98]	$X_{[1,t]}$	-	x_{t+c}, $c \geq 1$
Speech-XLNet [109]	$X_{\mathcal{P}_t}$		$x_{i \sim \mathcal{P}_t^c}$
DeCoAR [99]	$X_{[1,t-1]}$, $X_{[t+k+1,T]}$	-	$X_{[t,t+k]}$
Mockingjay [100], Audio ALBERT [119], DeCoAR 2.0 [110]	$X_{-[t,t+k]}$		$X_{[t,t+k]}$
TERA [106], BMR [103]	$X^{-[f,f+j]}_{-[t,t+k]}$		X
pMPC [105]	$X_{-[t,t+k']}$ ($X_{[t,t+k']}$ is a phoneme)		$X_{[t,t+k']}$
MPE [102]	X	$Z_{-[t,t+k]}$	Z
NPC [107]	X	$Z_{-[t,t+k]}$	X
	CONTRASTIVE MODELS		
Unspeech [120]	$X_{[t_1,t_2]}$	-	$X_{[t_0,t_1]}$,$X_{[t_2,t_3]}$
CPC [56], wav2vec [121], Modified CPC [122]	$X_{[1,t]}$	-	z_{t+c}, $c \geq 1$
Bidirectional CPC [123]	$X_{[1,t]}$ or $X_{[t,T]}$	-	z_{t+c} or z_{t-c}, $c \geq 1$
vq-wav2vec [124]	$X_{[1,t]}$	-	q_{t+c}, $c \geq 1$
wav2vec 2.0 [125], wav2vec-C [126][2]	X	$Z_{-[t,t+k]}$	$Q_{[t,t+k]}$
w2v-BERT [127]	X	$Z_{-[t,t+k]}$	$Q_{[t,t+k]}$ and $C_{[t,t+k]}$
Speech SimCLR [128][3]	AUGMENT$_1(X)$ and AUGMENT$_2(X)$		AUGMENT$_2(Z)$ and AUGMENT$_1(Z)$
	PREDICTIVE MODELS		
Discrete BERT [124], [129] [4]	$C_{-[t,t+k]}$		$C_{[t,t+k]}$
HuBERT [130][5], WavLM [131][6]	X	$Z_{-[t,t+k]}$	$C_{[t,t+k]}$
data2vec [132]	X	$Z_{-[t,t+k]}$	$\sum_l \bar{H}^{(l)}_{[t,t+k]}$
BEST-RQ [133][7]	$X_{-[t,t+k]}$		$C_{[t,t+k]}$

Subscripts indicate different augmentations. *Z* represents a localized latent representation sequence of *X*. $Z^{(l)}$ is Z at the *l*-th layer of the model used to compute it. $\bar{H}$ is the contextualized sequence *H* obtained from an exponential moving average (EMA) of the model undergoing training with no masking applied. *Q* represents a sequence of quantized learned representations, and *C* is a sequence of discrete cluster IDs. For contrastive models, we specify only positive targets.

[1]PASE uses multiple pretext tasks, but the authors find that reconstruction is most important.

[2]wav2vec-C adds reconstruction loss to wav2vec 2.0.

[3]Speech SimCLR targets the latent representation of an augmented version of X using a differently augmented X, and vice-versa.

[4]Discrete BERT obtains codes C from vq-wav2vec.

[5]HuBERT is trained first using cluster IDs of the MFCCs as target and subsequently clusters IDs of the model representations from the last iteration.

[6]WavLM simulates noisy/overlapped speech as inputs.

[7]BEST-RQ obtains codes C by quantizing acoustic features using a random projection quantizer.

representations may take the form of one-hot vectors, or the equivalent integer IDs, which we denote by $C = \{c_1, c_2, \ldots, c_T\}$. We use a circumflex to denote that, for instance, $\hat{x}_t$ is an approximation of x_t. Finally, we often use a subscript when defining a loss, $\mathcal{L}_i$, to imply that the total loss is computed as a sum over i, unless otherwise stated.

For some models, we will refer to H as a *contextualized* representation which means that each h_t is a function of some, linguistically speaking, long sub-sequence of X spanning at least several phonemes. Usually, h_t depends on the entire input X or all previous timesteps $X_{[1,t]}$. In contrast, a *localized* representation is one that only depends on a short part of the input $X_{[t-u,t+u]}$, where $u \geq 0$. The distinction between contextualized and localized may become fuzzy if u is large, however, this is rarely the case.

After pre-training, the representation model $f(\cdot)$ can be fine-tuned for a downstream task directly or used to extract features which are fed to another model, as visualized in Fig. 1. It is not uncommon to use the output representation H, but often representations from hidden layers of $f(\cdot)$ are better suited [68].

B. Generative Approaches

1) Motivation: In this category, the pretext task is to generate, or reconstruct, the input data based on some limited view. This includes predicting future inputs from past inputs, masked from unmasked, or the original from some other corrupted view. "Generative" as used in this paper hence refers to models that target the original input in their pretext task. Note that this differs from generative models, which learn distributions that allow to sample new data.

2) Approaches:

a) Autoencoding: Since their introduction in the mid-1990s [13], autoencoders (AEs) have played an essential role in learning distributed latent representations of sensory data. As described above, AEs consist of an encoder and decoder; the pretext task is to reconstruct the given input. The most common type of AE places an information bottleneck on the latent representation by simply having fewer hidden units available than input features. This forces the model to discard low-level details and disccourages the learning of trivial solutions. Other models add regularization to the latent space to further improve the quality of the learned representations. For instance, denoising autoencoders (DAEs) learn latent representations by reconstructing from input corrupted by noise [36]. The Variational Autoencoder (VAE) is a probabilistic version of the AE which defines the latent representation via a posterior distribution over stochastic latent variables [48], [49]. VAEs have been applied to speech in numerous works [70], [71], [72], [73], [74]. The vector-quantized variational autoencoder (VQ-VAE) is another model in this category [75]; it extends the original VAE [48] with a novel parameterization of the posterior distribution for discrete latent representations. The VQ-VAE has been instrumental in generative speech modelling and recent work on generative spoken language modeling has successfully combined the idea of a discrete latent space with self-supervised learning [76], [77], [78].

Specifically, in the VQ-VAE, the continuous representation vector h_t at the output of the encoder is quantized by mapping it to a codebook vector, which is then used as the input to the decoder. This operation is non-differentiable and the gradients of the loss with respect to the encoder parameters must be obtained by approximation. In the VQ-VAE this is done using the straight-through estimator [79], i.e., the gradients with respect to the encoder output are taken to be equal to those with respect to the decoder input (i.e., the quatization step is ignored). Given a learned codebook $A \in \mathbb{R}^{K \times D}$, where K is the codebook size and D is the dimensionality of each codebook vector a_k, the quantized representation q_t of h_t is obtained as

$$q_t = a_k, \text{ where } k = \arg\min_j \|h_t - a_j\|_2 . \tag{1}$$

The decoder $g(\cdot)$ is an autoregressive model that takes $q_{[1,t]}$ as input to generate x_t [80]. Codebook learning is facilitated by a two-term auxiliary loss similar to classical vector quantization dictionary learning [81], [82]. Gradients for the codebook vectors are given solely by a term that moves codebook vectors a_k closer to the non-quantized vectors h_t. A so-called *commitment term* is added to ensure that non-quantized vectors do not grow unboundedly by enforcing the encoder to keep them close to a codebook vector. This commitment term is optimized only by the encoder. The total VQ-VAE loss for a single timestep is

$$\mathcal{L}_t = \underbrace{\log p(x_t|q_{[1,t]})}_{\text{encoder+decoder}} + \underbrace{\text{MSE}\left(\text{sg}\left[h_t\right], A\right)}_{\text{codebook}} + \underbrace{\alpha\, \text{MSE}\left(h_t, \text{sg}\left[A\right]\right)}_{\text{encoder}}, \tag{2}$$

where $\log p(x_t|q_{[1,t]})$ is a reconstruction likelihood term usually using a categorical distribution, $\text{sg}[x] = x$ is the so-called stop-gradient operator which acts as the identity function during the forward pass but is assumed to have partial derivatives all equal to zero during the backward pass, α is a scalar hyperparameter, and we define $\text{MSE}(h_t, A) = \frac{1}{KD}\sum_{k=1}^{K}\sum_{i=1}^{D}(h_{t,i} - a_{k,i})^2$. The loss for a full sequence is the sum or mean over all $\mathcal{L}_t$.

These learned discrete representations have been shown to capture high-level speech information closely related to phonemes, and are useful for applications such as speaker conversion [83]. Vector quantization is not exclusive to VQ-VAE but has seen widespread application within SSL for regularization purposes and to define targets for the pretext task. We will cover these applications below.

The Gumbel softmax [84] is another frequently used approach for obtaining a discrete representation space, and has also been used for AEs [85]. In addition to the approaches discussed above, several other works on speech representation learning take inspiration from the AE framework [86], [87], [88], [89], [90], [91].

b) Autoregressive prediction: Autoregressive predictive coding (APC) [92], [93] takes inspiration from the classic Linear Predictive Coding (LPC) approach for speech feature extraction [94] and autoregressive language models (LM) for text, where the model learns to predict future information from past. A function $f(\cdot)$ reads the input sequence $X_{[1,t]}$ and outputs a

representation sequence $H_{[1,t]}$. The auxiliary module $g(\cdot)$ is a linear projection layer which takes the last vector of $H_{[1,t]}$ as input to approximate x_{t+c}, where $c \geq 1$. Thus, c indicates how many timesteps the model predicts ahead. The modules $f(\cdot)$ and $g(\cdot)$ are jointly learned to minimize the L_1 loss between x_{t+c} and its approximation $\hat{x}_{t+c}$. APC is formulated as

$$H_{[1,t]} = f(X_{[1,t]}), \tag{3}$$

$$\hat{x}_{t+c} = g(h_t), \tag{4}$$

$$\mathcal{L}_t = \|\hat{x}_{t+c} - x_{t+c}\|_1. \tag{5}$$

In text-based autoregressive LMs, c is set to 1 to enable autoregressive generation. However, due to the smoothness of the speech signal, neighboring acoustic features are usually similar. Depending on the downstream task, we are often interested in learning so-called *slow features* that typically span multiple input frames [95]. Even the smallest linguistic units of speech—phonemes—span 0.07 seconds on average in the English TIMIT dataset [96], whereas spectrogram frames $\mathbf{x}_t$ are typically computed at 0.01 s intervals. Thus, simply predicting the next frame constitutes a trivial pretext task for APC; the original work finds that $c = 3$ performs well. In [97], the APC objective is extended to multi-target training. The new objective generates both past and future frames conditioned on previous context. In VQ-APC [98], quantization is used with the APC objective, which imposes an information bottleneck serving as a regularizer.

A drawback of APC is that it encodes information only from previous timesteps and not the entire input. DeCoAR [99] combines the bidirectionality of the popular NLP model ELMo [51] and the reconstruction objective of APC to alleviate this issue and allow encoding information from the entire input. It uses a forward LSTM $f_1(\cdot)$ to encode $X_{[1,t]}$ and a backward LSTM $f_2(\cdot)$ to encode $X_{[t+k,T]}$, where $k > 1$:

$$H_{[1,t]} = f_1(X_{[1,t]}), \tag{6}$$

$$H'_{[t+k,T]} = f_2(X_{[t+k,T]}), \tag{7}$$

$$\hat{X}_{[t+1,t+k-1]} = g(h_t, h'_{t+k}). \tag{8}$$

The input feature vector used in the downstream tasks is the concatenation of h_t and h'_t.

c) Masked Reconstruction: Masked reconstruction is largely inspired by the masked language model (MLM) task from BERT [54]. During BERT pre-training, some tokens in the input sentences are masked by randomly replacing them by a learned masking token or another input token. The model learns to reconstruct the masked tokens from the non-masked tokens. Recent work has explored similar pretext tasks for speech representation learning. Similar to the DeCoAR model described above, this allows a model to learn contextualized representations that encode information from the entire input. While we here focus on the models that reconstruct the masked input, it is important to note that masking has also been used extensively for contrastive (Section III-C) and predictive (Section III-D) models.

From a high-level perspective, the training phase of models using masked reconstruction can be formulated as

$$H = f(m(X)), \tag{9}$$

$$\hat{x}_t = g(h_t), \tag{10}$$

$$\mathcal{L}_t = \|\hat{x}_t - x_t\|_1. \tag{11}$$

The exact masking policy defined by $m(\cdot)$ differs from model to model and will be discussed further below. The function $f(\cdot)$ is typically a Transformer encoder [100], [101], [102], but recurrent neural networks have also been used [103]. In general, the Transformer encoder architecture has been adopted widely by self-supervised models for speech within all three surveyed categories. The function $g(\cdot)$ is usually a linear projection or a multilayer perceptron (MLP). Finally, the loss $\mathcal{L}_t$ is commonly computed only for masked timesteps in order to discourage the model from learning an identity mapping.

The masking policies used in NLP can be adapted to speech by considering a speech segment equivalent to a token in a sentence; indeed, the masking strategy of BERT has also been used for speech pre-training [100]. In the standard BERT masking policy, each token is masked independently at random. However, for speech, masking a single sample or spectrogram frame results in a largely trivial reconstruction task since, as discussed in paragraph III-B2b, the smoothness of audio signals may encourage the model to learn to simply interpolate neighboring frames. Therefore it is common to mask chunks of consecutive frames [100], [104].

We can bring the pretext task closer to the NLP equivalent by using a masking policy where the masked regions of the input correspond to linguistic units. Instead of just masking a fixed number of consecutive frames, pMPC [105] selects masked speech frames according to the phonetic segmentation in an utterance. However, in order to obtain this segmentation, some labeled data is of course needed.

Whereas most studies use masking along the temporal dimension of the input, speech can also be masked along the frequency dimension when spectral input features are used [103], [106]. Frequency masking has been shown to improve representations used for speaker classification [106].

Some studies explore alternatives to masking the input directly. In non-autoregressive predictive coding (NPC) [107], time masking is introduced through masked convolution blocks. Taking inspiration from XLNet [108], it has also been suggested that the input be reconstructed from a shuffled version [109] to address the discrepancy between pre-training and fine-tuning of masking-based approaches.

Regularization methods can further improve on masked reconstruction approaches. DeCoAR 2.0 [110] uses vector quantization, which is shown to improve the learned representations. Furthermore, two dropout regularization methods—attention dropout and layer dropout—are introduced with the TERA model [106], [111]. Both methods are variations on the original dropout method [112].

d) More Generative Approaches: Other than the autoregressive and masked reconstruction tasks discussed above, various studies have explored the reconstruction of other targets

derived from the input. PASE and PASE+ [69], [113] use multiple targets, including the waveform, log power spectrum, mel cepstral coefficients (MFCCs), and prosody features. Models that learn acoustic embeddings of small speech segments have targeted future and past spectrogram segments [114], [115], [116], phase information [117], and the temporal gap between two segments [115], [116].

3) Challenges: Although successful NLP models like BERT and GPT are based on generative pretext tasks, the progress have not been translated directly to the speech domain. A speech signal encodes more information than text, such as speaker identity and prosodic features, which makes it harder to generate. However, in order to generate all details of the input, the model must encode all information in the speech signal. Hence, a model that learns to perfectly reconstruct its input may not necessarily have learned to isolate the features of interest and will encode redundant information for a given downstream task.

There are many choices involved in designing a generative pretext task. For instance, masking strategy and the choice of input and target representation (e.g., waveform samples or spectral features). These choices influence what the model learns through the pretext task. However, there is little research on the relationship between task design and the information encoded in the learned representations.

C. Contrastive Approaches

1) Motivation: As discussed above, speech contains many entangled features. Thus, learning to reconstruct the raw speech signal might not be the best way to discover contextualized latent factors of variations. Contrastive models learn representations by distinguishing a target sample (positive) from distractor samples (negatives) given an *anchor representation*. The pretext task is to maximize latent space similarity between the anchor and positive samples while minimizing the similarity between the anchor and negative samples. This approach has been used extensively in the general ML community [134].

2) Approaches:

a) CPC: Contrastive Predictive Coding (CPC) [56] is a prominent example of a contrastive model. CPC uses a convolutional module $f_1(\cdot)$ to produce localized representations z_t with a recurrent module $f_2(\cdot)$ on top that outputs a contextualized representation h_t. An anchor representation $\hat{z}_{t,k}$ is obtained via a linear projection $g_k(\cdot)$ of h_t. The positives and negatives are sampled from the localized representation Z. Hence, at a single timestep t, CPC forms multiple anchor representations $\hat{z}_{t,k}$ for $k \in \{1, \ldots, K\}$ and associates with each one a single positive sample at the corresponding timestep, z_{t+k}, k steps in the future:

$$z_t = f_1(X_{[t-u,t+u]}), \tag{12}$$

$$H_{[1,t]} = f_2(Z_{[1,t]}), \tag{13}$$

$$\hat{z}_{t,k} = g_k(h_t). \tag{14}$$

Each z_t only encodes information from a limited receptive field, while $f_2(\cdot)$ is limited to condition each h_t on previous timesteps $Z_{[1,t]}$. Without these restrictions, the model could collapse to a trivial solution. g_k is a unique transformation per offset k (e.g., a linear projection). The loss function measures the similarity between the anchor representation $\hat{z}_{t,k}$ and the positive z_{t+k} normalized by the total similarity to the positive and negatives. The approach is similar to previous work on Noise-Contrastive Estimation (NCE) [135]. Minimizing the loss corresponds to maximizing a lower bound on the mutual information between h_t and z_{t+k} (and in turn $x_{t+k-u:t+k+u}$) and is hence called InfoNCE:

$$\mathcal{L}_{t,k} = -\log\left(\frac{\exp(\hat{z}_{t,k}^{\mathrm{T}} z_{t+k})}{\sum_{i\in\mathcal{I}} \exp(\hat{z}_{t,k}^{\mathrm{T}} z_i)}\right). \tag{15}$$

Here, $\mathcal{I}$ is a random subset of N indices which includes the target index $t+k$ and $N-1$ negative samples drawn from a proposal distribution, e.g., a uniform distribution over $\{1, \ldots, T\}$. Including the target index in $\mathcal{I}$ ensures that the loss is a proper categorical cross-entropy and that minimizing it has the previously stated relation to mutual information maximization. This corresponds to sampling negatives from the same sequence and has been shown to give good performance for phoneme classification [56]. The loss is indexed by k to show that CPC targets multiple offsets using different projection layers $g_k(\cdot)$. The authors find $K = 12$ to work well for phoneme classification.

The wav2vec model [121] extends the CPC approach and uses fully convolutional parameterizations for the modules $f_1(\cdot)$ and $f_2(\cdot)$ with receptive fields of $30\,\mathrm{ms}$ and $210\,\mathrm{ms}$, respectively. While the CPC loss solves a 1-of-N classification task per (t, k), either assigning the anchor to the positive class or (wrongly) to one of the $N-1$ negative classes, the wav2vec loss considers a sequence of N independent binary classifications. That is, the anchor is compared independently to the positive and each negative, and the loss is computed as a sum of the associated log-probabilities,

$$\mathcal{L}_{t,k} = -\log\left(\sigma\left(\hat{z}_{t,k}^{\mathrm{T}} z_{t+k}\right)\right) + \sum_{i\in\mathcal{I}} \log(1 - \sigma(\hat{z}_{t,k}^{\mathrm{T}} z_i)). \tag{16}$$

Here, $\sigma(x) = 1/(1+\exp(-x))$ is the sigmoid function, $\sigma(\hat{z}_{t,k}^{\mathrm{T}} z_{t+k})$ is the probability of the anchor being the positive sample and $\sigma(\hat{z}_{t,k}^{\mathrm{T}} z_i)$ is the probability of the anchor being the negative sample. Evidently and contrary to CPC, $\mathcal{I}$ must not include the target index $t+k$ as this would cancel out the positive term.

b) wav2vec 2.0: The wav2vec 2.0 model combines contrastive learning with masking. As the CPC model, it uses the InfoNCE loss [56] to maximize the similarity between a contextualized representation and a localized representation. However, instead of using the z_t directly as positive and negatives, it uses a quantization module $g(\cdot)$ to obtain a discrete representation. This has the practical implication that one can avoid sampling negatives from the same category as the positive. The model takes as input a waveform and uses a convolutional module $f_1(\cdot)$ followed by a Transformer encoder $f_2(\cdot)$. Masking is applied to the output of the convolutional module:

$$z_t = f_1(X_{[t-u,t+u]}), \tag{17}$$

$$H = f_2(m(Z)), \tag{18}$$

$$q_t = g(z_t). \tag{19}$$

The quantization module $g(\cdot)$ uses a Gumbel softmax [84] with a straight-through estimator. Since the quality of the learned representations is contingent on the quality of the quantization, wav2vec 2.0 combines two techniques to learn high-quality codebooks. First, wav2vec 2.0 concatenates quantized representations from multiple codebooks at each timestep, so-called Product Quantization (PQ) [136]. Also, the primary training loss described below is augmented with an auxiliary term designed to encourage equal use of all codebook entries.

In wav2vec 2.0, anchors are taken to be h_t at masked timesteps only, the positive sample is chosen as the quantized vector, q_t, at the same timestep, and negatives are sampled from other masked timesteps. The loss is

$$\mathcal{L}_t = -\log\left(\frac{\exp(S_{\mathrm{c}}(h_t, q_t))}{\sum_{i\in\mathcal{I}}\exp(S_{\mathrm{c}}(h_t, q_i))}\right), \tag{20}$$

where $S_{\mathrm{c}}(\cdot)$ is the cosine similarity and $\mathcal{I}$ contains the target index t and negative indices sampled from other masked timesteps.

The wav2vec 2.0 approach was the first to reach single-digit word error rate (WER) on LibriSpeech using only the low-resource Libri-light subsets for fine-tuning a pre-trained model (see Section IV-B). It has subsequently inspired many follow-up studies. The wav2vec-C [126] approach extends wav2vec 2.0 with a consistency term in the loss that aims to reconstruct the input features from the learned quantized representations, similar to VQ-VAE [137].

3) Challenges: Although representations learned using contrastive approaches have proved effective across a wide range of downstream applications, they face many challenges when applied to speech data. One challenging aspect is that the strategy used to define positive and negative samples can also indirectly impose invariances on the learned representations. For example, sampling negatives exclusively from the same utterance as the positive biases the features towards speaker invariance, which may or may not be desired for downstream applications. Another standing challenge is that since speech input does not have explicit segmentation of acoustic units, the negative and positive samples do not represent a whole unit of language but rather partial or multiple units, depending on the span covered by each sample. Finally, since speech input is smooth and lacks natural segmentation, it can be difficult to define a contrastive sampling strategy that is guaranteed to provide samples that always relate to the anchor as truly positives and negatives in a sound way.

D. Predictive Approaches

1) Motivation: Similar to the contrastive approaches discussed above, predictive approaches are defined by using a learned target for the pretext task. However, unlike the contrastive approaches, they do not employ a contrastive loss and instead use a loss function such as squared error and cross-entropy. Whereas a contrastive loss discourages the model from learning a trivial solution by the use of negative samples, this must be circumvented differently for predictive methods. For this reason, predictive methods compute the targets outside the model's computational graph; usually with a completely separate model. Thus, the predictive setup is somewhat akin to teacher-student training. The first predictive approaches were motivated by the success of BERT-like methods in NLP [54] as well as the DeepCluster method in CV [138].

2) Approaches:

a) Discrete BERT: Applying BERT-type training directly to speech input is not possible due to its continuous nature. The Discrete BERT approach [129] uses a pre-trained vq-wav2vec model to derive a discrete vocabulary [124]. The vq-wav2vec model is similar to wav2vec mentioned in Paragraph III-C2a but uses quantization to learn discrete representations. Specifically, discrete units c_t are first extracted with the vq-wav2vec model $f_1(\cdot)$ and then used as inputs *and* targets in a standard BERT model $f_2(\cdot)$ with a softmax normalized output layer $g(\cdot)$,

$$c_t = f_1(X_{[t-u,t+u]}), \tag{21}$$

$$H = f_2(m(C)), \tag{22}$$

$$\hat{c}_t = g(h_t). \tag{23}$$

Similar to BERT, the model can then be trained with a categorical cross-entropy loss,

$$\mathcal{L} = \sum_{t\in\mathcal{M}} -\log p(c_t \mid X), \tag{24}$$

where $\mathcal{M}$ is the set of all masked timesteps. During training, only the BERT model's parameters are updated, while the vq-wav2vec model parameters are frozen. Discrete BERT was the first model to demonstrate the effectiveness of self-supervised speech representation learning by achieving a WER of 25% on the standard test-other subset using a 10-minute fine-tuning set, setting the direction for many approaches to follow.

b) HuBERT: Rather than relying on an advanced representation learning model for discretizing continuous inputs, as Discrete BERT, the Hidden Unit BERT (HuBERT) approach [130] uses quantized MFCC features as targets learned with classic k-means. Thus, to compute the targets, the k-means model $g_1(\cdot)$ assigns a cluster center to each timestep. Different from Discrete BERT, HuBERT takes the raw waveform as input, rather than discrete units. This helps to prevent loss of any relevant information due to input quantization. HuBERT uses an architecture similar to that of wav2vec 2.0, with a convolutional module $f_1(\cdot)$ and a Transformer encoder $f_2(\cdot)$, as well as a softmax normalized output layer $g_2(\cdot)$:

$$c_t = g_1(X_{[t-w,t+w]}), \tag{25}$$

$$z_t = f_1(X_{[t-u,t+u]}), \tag{26}$$

$$H = f_2(m(Z)), \tag{27}$$

$$\hat{c}_t = g_2(h_t), \tag{28}$$

where w defines the window size used to compute the MFCCs. The categorical cross-entropy loss is computed on both masked, $\mathcal{L}_m$, and unmasked, $\mathcal{L}_u$, timesteps:

$$\mathcal{L}_m = \sum_{t\in\mathcal{M}} -\log p(c_t \mid X), \tag{29}$$

$$\mathcal{L} = \beta\mathcal{L}_m + (1-\beta)\mathcal{L}_u. \tag{30}$$

Again, $\mathcal{M}$ is the set of all masked timesteps, β is a scalar hyperparameter and $\mathcal{L}_u$ is computed as $\mathcal{L}_m$ but summing over $t \notin \mathcal{M}$.

Intuitively, the HuBERT model is forced to learn both an acoustic and a language model. First, the model needs to learn a meaningful continuous latent representation for unmasked timesteps which are mapped to discrete units, similar to a classical frame-based acoustic modeling problem. Second, similar to other masked pre-training approaches, the model needs to capture long-range temporal dependencies to make correct predictions for masked timesteps.

One crucial insight motivating this work is the importance of consistency of the targets which enables the model to focus on modeling the sequential structure of the input. Importantly though, for HuBERT, pre-training is a two-step procedure. The first iteration is described above. Once completed, a second iteration of pre-training follows. Here, representations from a hidden layer of the model from the first iteration are clustered with k-means to obtain new targets c_t.

For HuBERT, only two iterations are needed to match or outperform the previous state-of-the-art results for low-resource speech recognition. And combining the HuBERT approach with the wav2vec 2.0 approach, the w2v-BERT model has managed to improve results even further [127].

c) WavLM: WavLM emphasizes spoken content modeling and speaker identity preservation [131]. It is largely identical to HuBERT, but introduces two useful extensions.

First, it extends the Transformer self-attention mechanism with a so-called *gated relative position bias*. The bias is added prior to the softmax normalization of the attention weights. For the attention weight at i, j, the bias is computed based on the input to the Transformer layer at the current time step i and also incorporates a relative positional embedding for $i - j$. The authors find that this extension improves performance on phoneme and speech recognition tasks.

Second, it uses an utterance mixing strategy where signals from different speakers are combined to augment the training data. Specifically, random subsequences from other examples in the same batch are scaled and added to each input example. Only the targets corresponding to the original example are predicted during pre-training. Thus, the model learns to filter out the added overlapping speech.

Most SSL methods are trained on data where each example only contains speech from a single person; therefore, they can perform subpar on multispeaker tasks like speaker separation and diarization.

The WavLM model achieved substantial improvements on the speech separation, speaker verification and speaker diarization tasks in the SUPERB benchmark, while also performing well on many other tasks compared to HuBERT and wav2vec 2.0.

d) data2vec: Motivated by the success of using an exponential moving average (EMA) teacher for self-supervised visual representations [139], [140], the data2vec model [132] computes targets Y using an EMA of its own parameters. The targets are constructed by averaging hidden representations of the top k layers of the EMA teacher network applied to unmasked inputs. Here, we denote this jointly as $\bar{f}_2(\cdot)$.

The data2vec model was proposed for different data modalities, but for audio it uses an architecture similar to wav2vec 2.0 and HuBERT with a convolutional module $f_1(\cdot)$, a Transformer $f_2(\cdot)$ and masking applied to the Transformer input.

$$z_t = f_1(X_{[t-u,t+u]}), \tag{31}$$

$$H = f_2(m(Z)), \tag{32}$$

$$Y = \bar{f}_2(Z). \tag{33}$$

The teacher network $\bar{f}_2(\cdot)$ is a copy of the Transformer of the student network but with the parameters at training step i, $\theta_{\text{teacher},i}$, given by an EMA of the student parameters over all previous training steps.

$$\theta_{\text{teacher},i} = \begin{cases} \theta_{\text{student},0} & i = 0 \\ \gamma\theta_{\text{student},i} + (1-\gamma)\theta_{\text{teacher},i-1} & i > 0 \end{cases}, \tag{34}$$

where $\theta_{\text{student},i}$ are the parameters of the student network at training step i, updated via gradient descent, and γ is the EMA decay rate.

The data2vec model uses a regression loss between target and prediction. Specifically, to reduce sensitivity to outliers and prevent exploding gradients, it uses the smoothed L_1 loss [141],

$$\mathcal{L}_t = \begin{cases} \frac{1}{2}(y_t - h_t)^2/\eta, & |y_t - h_t| \le \eta \\ |y_t - h_t| - \frac{1}{2}\eta, & \text{otherwise} \end{cases}, \tag{35}$$

where the hyperparameter η controls the transition from a squared loss to an L_1 loss.

The data2vec approach was shown to work well for representation learning with either speech, images or text data. It is the first approach to achieve competitive results when trained on any one of the three modalities.

3) Challenges: The iterative nature of pre-training for the HuBERT and wavLM could present a practical inconvenience when working with large volumes of data. Another challenge for these models centers around the quality of the initial vocabulary generated from MFCC features. The data2vec approach improves over other predictive models by allowing the targets to improve continuously via the EMA teacher network; however, student-teacher approaches inflate the existing computational challenges of very large models and may necessitate the use of methods that decrease instantaneous memory utilization such as mixed precision training, model parallelism and model sharding [142].

E. Learning From Multi-Modal Data

1) Motivation: Multiple modalities are useful in many settings, where each modality provides information that is complementary to other modalities. Multi-modal work includes supervised settings, such as audio-visual ASR [143], [144] and person identification [145] which have been studied for decades. In this section, we focus only on unsupervised representation learning from multi-modal data.

One of the motivations for learning from multiple modalities is that it can reduce the effect of noise, since noise in different modalities is likely to be largely independent or uncorrelated. In addition, learning from speech data with accompanying signals such as images or video can help learn representations that encode more semantic information. Such "grounding" signals can contain supplementary information that can be used by models to infer the content of the speech. Human language learning provides a proof of concept for this, as it is believed that infants benefit from the visual modality when learning language [146]. Early computational models of multi-modal language learning were motivated by (and tried to emulate) human learning of language in the context of the visual surroundings [147].

2) Approaches: We define two broad classes of approaches in this area. Specifically, depending on what type of multi-modal data is involved we refer to "intrinsic" or "extrinsic" modalities.

Intrinsic modalities are modalities produced directly by the speech source. Examples of intrinsic modalities (besides the speech audio) include images or video of the speaker's face [148], [149], lip-movement [150], articulatory flesh point measurements [151], [152], or simultaneous magnetic resonance imaging (MRI) scans [153]. Typically, learning from multiple intrinsic modalities is done so as to improve robustness to noise, since acoustic noise is likely to be uncorrelated with the other modality(ies). This type of representation learning is often referred to as "multi-view learning" because the multiple intrinsic modalities can be seen as multiple views of the same content. Some typical approaches in this category include

- Multi-view autoencoders and variations [154], [155],
- Multi-modal deep Boltzmann machines [156],
- Canonical correlation analysis (CCA) [157] and its nonlinear extensions [158], [159], [160], [161], [162], [163], [164], [165], [166],
- Multi-view contrastive losses [167], [168],
- More recently, audio-visual extensions of masked prediction methods [150], [169], specifically Audio-Visual HuBERT (AV-HuBERT) [150].

Extrinsic modalities are modalities that are not produced by the same source but still provide context for each other. A typical example is an image and its spoken caption: The image tells us what the speech is likely describing, so a representation model that takes both modalities into account will hopefully encode more of the meaning of the speech than a single-modality model. There has recently been a surge of datasets collected for this purpose, usually consisting of images and spoken captions, the audio and image frames in a video, or video clips with their spoken descriptions. A recent review of datasets, as well as methods, in this category is provided by Chrupała [170].

Typical approaches involve learning a neural representation model for each modality, with a multi-modal contrastive loss that encourages paired examples in the two modalities to have similar representations while unpaired examples remain different [171], [172], [173], [174], [175], [176]. Other choices include training with a masked margin softmax loss [177], [178] or a masked prediction loss [179]. Such models are typically evaluated on cross-modal retrieval, although some work has also used the models for other downstream tasks such as the ZeroSpeech and SUPERB benchmark tasks [180]. Analyses of such models have found that, despite the very high-level learning objective of matching speech with a corresponding image (or other contextual modality), such models often learn multiple levels of linguistic representations from the shallowest to the deepest model layers [181], [182], [183]. They are also able to learn word-like units [184], [185], [186] and can be used for cross-lingual retrieval, by considering the visual signal as an "interlingua" [187], [188], [189]. In some settings, even in the presence of some amount of textual supervision (i.e., the speech is transcribed), visual grounding still helps learn a better representation for retrieval [190].

There has also been growing interest in learning joint speech and text representations using paired and unpaired data. The SLAM approach [191] is an example where speech and text are first represented using two separate pre-trained encoders followed by a multi-modal encoder to build the joint representations. The entire model is trained using a multi-task loss including two supervised and two self-supervised tasks.

3) Challenges: One of the challenges of using multi-modal approaches is that the multi-modal data they rely on is often in shorter supply than single-modality data. In addition, multi-modal data is typically drawn from specific domains, for example domains involving descriptions of visual scenes. It is not clear how well the learned speech representations apply to other speech domains that are not necessarily describing or situated in a visual scene, and this question requires further study.

F. Acoustic Word Embeddings

Most of the representation learning techniques discussed in the preceding sections are aimed at learning frame-level representations. For some purposes, however, it may be useful to explicitly represent longer spans of speech audio of arbitrary duration, such as phone, word, or phrase-level segments. For example, searching within a corpus of recorded speech for segments that match a given (written or spoken) query can be seen as finding segments whose representations are most similar to that of the query [118], [192], [193], [194]; word embeddings can be defined by pooling representations of instances of a given word [114]; unsupervised segmentation and spoken term discovery can be seen as a problem of detecting and clustering segments [195], [196]; and even ASR can be viewed as the problem of matching written word representations to representations of audio spans [91], [197], [198].

Several lines of work have begun to address the problem of learning representations of spans of speech, especially word segments, typically referred to as *acoustic word embeddings*. Early work on unsupervised acoustic word embeddings defined them as vectors of distances from the target segment to a number of pre-defined "template" segments [199]. Later work used variants of neural autoencoders [118], [200], [201], [202].

TABLE II
SUMMARY OF DATASETS USED IN PRE-TRAINING (DENOTED AS PT) OR EVALUATING (DENOTED AS EV) SSL TECHNIQUES IN THE LITERATURE. THE LANGUAGES AND SIZES OF THE DATASETS ARE PROVIDED IN COLUMNS 3 AND 4. COLUMN 5 LISTS THE TASKS EACH DATASET IS USED TO EVALUATE

Dataset	Purpose	Lang.	Size [hours]	Task	License
LibriLight (LL)	PT	EN	60k	-	MIT License
AudioSet	PT	Multi	2.5k	-	CC BY 4.0
AVSpeech	PT	Multi	3.1k	-	CC BY 4.0
Fisher	PT	EN	2k/1k [104]	-	Linguistic Data Consortium (LDC)
Alexa-10k	PT	EN	10k	-	Not released
Didi Callcenter	PT	ZH	10k	-	Not released
Didi Dictation	PT	ZH	10k	-	Not released
LibriSpeech (LS)	PT/EV	EN	960	ASR/PR/PC/SID	CC BY 4.0
Wall Street Journal (WSJ)	PT/EV	EN	81	ASR/PR/PC/SID	Linguistic Data Consortium (LDC)
Common Voice (CV-dataset)	PT/EV	Multi	11k/7k [218]/430 [123]	ASR/PR/PC	CC0
Multilingual LS (MLS)	PT/EV	Multi	50k	ASR	CC BY 4.0
VoxPopuli (VP)	PT/EV	Multi	400k [218]/24k [131], [219]	ASR	CC0
BABEL (BBL)	PT/EV	Multi	1k	ASR	IARPA Babel Agreement
GigaSpeech	PT/EV	EN	40k/10k [131], [219]	ASR	Apache-2.0 License
TED-LIUM 3 (TED3)	PT/EV	EN	450	ASR	CC BY-NC-ND 3.0
TED-LIUM 2 (TED2)	PT/EV	EN	118	ASR	CC BY-NC-ND 3.0
Switchboard (SWB)	PT/EV	EN	260	ASR	Linguistic Data Consortium (LDC)
TIMIT	PT/EV	EN	4	ASR/PR/PC	Linguistic Data Consortium (LDC)
VoxLingua107	PT/EV	Multi	6.6k	LID	CC BY 4.0
Open Mandarin	PT/EV	ZH	1.5k	ASR	CC BY-NC-ND 4.0, Apache License v.2.0, Linguistic Data Consortium (LDC)
HKUST	PT/EV	ZH	168/200	ASR	Linguistic Data Consortium (LDC)
AISHELL-1	PT/EV	ZH	178	ASR	Apache License v.2.0
Hub5'00	EV	EN	13	ASR	Linguistic Data Consortium (LDC)
DIRHA	EV	EN	11	ASR	See link for details[8]
CHiME-5	EV	EN	50	ASR	See link for details[9]
Alexa-eval	EV	EN	1k	ASR	Not released
INTERFACE	EV	Multi	16	Sentiment	No information
MOSEI	EV	EN	65	Sentiment	See link for details[10]
VCTK	EV	EN	44	SID/ASV	CC BY 4.0
VoxCeleb1	EV	Multi	352	SID/ASV	CC BY 4.0
Fluent Speech Commands (FSC)	EV	EN	14.7	IC	CC BY-NC-ND 4.0
QUESST 2014 (QUESST)	EV	Multi	23	QbE	No information
LS En-Fr	EV	En-Fr	236	ST	CC BY 4.0
CoVoST-2	EV	Multi	2.9k	ST	CC0
ALFFA	EV	Multi	5.2–18.3	ASR-multi	MIT License
OpenSLR-multi	EV	Multi	4.4–265.9	ASR-multi	CC BY-SA 3.0 US, CC BY-SA 4.0, CC BY 4.0, CC BY-NC-ND 4.0, Apache License v.2.0
AED datasets	EV	-	-	AED	CC BY 4.0 (MUSAN, Speech Commands, NSynth, Bird Audio Detection), CC0 (Spoken Language Identification), Non-Commercial (TUT)

Abbreviations: **EN**: English; **MULTI**: Multilingual; **ZH**: Chinese; **FR**: French; **ASR**: Automatic speech recognition; **PR**: Phoneme Recognition; **PC**: Phoneme classification; **SID**: Speaker identification; **ASV**: Automatic speaker verification; **SENTIMENT**: Sentiment analysis; **ST**: Speech translation; **QBE**: Query by example or spoken term detection; **IC**: Intent classification; **AED**: Audio event detection; and **LID**: Language identification. We distinguish **PR** from **PC** based on whether the inference is made at the phone level sequentially or the frame level separately. **SID** and **ASV** both evaluate model capability in encoding speaker information; **SID** classifies one utterance into a pre-defined set of speaker labels, whereas **ASV** infers whether a given pair of utterances was uttered by the same speaker.

[8][Online]. Available: https://dirha.fbk.eu/node/107

[9][Online]. Available: https://chimechallenge.github.io/chime6/download. html

[10][Online]. Available: https://github.com/A2Zadeh/CMU-MultimodalSDK/blob/master/LICENSE.txt

These are often evaluated on word discrimination, that is the task of determining whether two word segments correspond to the same word or not [203]. This task can be thought of as a proxy for query-by-example search, since the basic operation in search is to determine whether a segment in the search database matches a query segment, and has been used for evaluation of both frame-level (e.g., [89]) and word-level [199], [204] representations.

Since most work on acoustic word embeddings preceded the very recent wave of new self-supervised frame-level representations, one question is whether word (or more generally segment) embeddings could be derived more simply by pooling self-supervised frame-level representations, as has been done for text span embeddings by pooling over word embeddings [205], [206]. Some initial results suggest that at least very simple pooling approaches like downsampling and mean or max pooling are not successful [202], [207], but more work is needed to reach conclusive results.

IV. BENCHMARKS FOR SELF-SUPERVISED LEARNING

The previous sections presented various methodologies by which to learn speech representations from unlabeled corpora. This section surveys the datasets available to learn and evaluate these representations. We also summarize several studies and their results to demonstrate the usefulness of the learned representations for various downstream tasks.

A. Datasets Only for Pre-Training

Table II summarizes datasets used for pre-training SSL techniques in the literature. These datasets are usually large but with limited or no labels. Libri-light (LL) [208], one of these datasets,

is derived from audiobooks that are part of the LibriVox[1] project. LL contains 60 k hours of spoken English audio tagged with SNR, speaker ID, and genre descriptions. The speech examples in Audioset [209], which consists of over 2 M 10-second YouTube video clips human-annotated with 632 audio events, have also been used for pre-training. Audioset has 2.5 k hours of audio of varying quality, different languages, and sometimes multiple sound sources. AVSpeech [210] is another large-scale audio-visual dataset used in SSL research, comprising 4.7 k hours of clips from a wide variety of languages. Each clip contains a visible face and audible sound originating from a single speaker without interfering background signals. The 3100-hour audio part of AVSpeech has been used to learn audio-only representations [123]. The Fisher corpus [211] collects over 2 k hours of conversational telephone speech, 1 k hours of which is utilized for pre-training [104]. Industrial researchers have also begun to build large-scale datasets for learning speech representations. For instance, 10 k hours of real-world far-field English voice commands for self-supervised pre-training have been collected at Amazon [126].

In addition to these English and multilingual efforts, researchers have also collected corpora for pre-training Chinese speech representations. Didi Dictation and Didi Call-center [101], [104] are two internal datasets containing respectively 10 k hours of read speech collected from mobile dictation application and 10 k hours of spontaneous phone calls between users and customer service staff.

B. Datasets for Both Pre-Training and Evaluation

Several datasets that provide both speech and associated transcripts and speaker labels have also been used to develop SSL techniques by enabling in-domain pre-training and evaluation. Such datasets are also listed in Table II. One of the most commonly used datasets in this category is LibriSpeech (LS) [212], a labeled corpus containing 960 hours of read English speech, which is also derived from an open-source audiobooks project[8]. The corpus consists of subsets *train-clean-100*, *train-clean-360*, *train-other-500*, *dev-clean*, *dev-other*, *test-clean*, and *test-other* used for training, development, and testing, respectively. Subsets tagged with *other* are more challenging utterances from speakers that yield higher WER as measured with previously built models. LS is used for unsupervised representation pre-training by ignoring its labels, and can also be utilized to evaluate the performance of representation on ASR, phoneme recognition (PR), phoneme classification (PC), and speaker identification (SID) tasks. Wall Street Journal (WSJ) [213] is another widely adopted, labeled corpus for pre-training. Its labels can evaluate performance for ASR, PR, PC, and SID. The original WSJ corpus contains 400 hours of English read speech data, and today its *si284* (81 hours), *dev93*, *eval92* subsets are the most-used partitions for unsupervised training, development, and test, respectively. The *si84* (15 hours) partition is also used for training.

The speech community also utilizes multilingual corpora. These are often large-scale, which facilitates pre-training, but are also partially labeled for ASR evaluation (PC and PR can be enabled via phone-level forced alignment). These corpora include Common Voice (CV) [214], Multilingual LibriSpeech (MLS) [215], VoxPopuli (VP) [216], and BABEL (BBL) [217]. CV is an open-source, multi-language, growing dataset of voices containing 11 k hours of audio from 76 languages as of the date this review was written (Common Voice corpus 7.0). Researchers usually use part of this for pre-training (e.g., 7k hours/60 languages in [218] and 430 hours/29 languages in [123]) or evaluation. MLS derives content from read audiobooks of LibriVox and contains data in eight European languages for a total of 50 k hours of audio. VP comprises a total of 400k hours of parliamentary speech from the European parliament in 23 European languages. The entire dataset [218] or a 24k-hour portion [131], [219] thereof has been used for pre-training. BBL consists of 1 k hours of conversational telephone speech in 17 African and Asian languages.

Several datasets, including GigaSpeech [220], TED-LIUM 3 (TED3) [221], TED-LIUM 2 (TED2) [222], Switchboard (SWB) [223], TIMIT [96], and VoxLingua107 [224], are labeled and conventionally used for evaluation, while their audio streams are also aggregated to build diversified and large-scale corpora for unsupervised pre-training [109], [123], [218]. GigaSpeech is a multi-domain English ASR corpus with 33 k hours of audio collected from audiobooks, podcasts, and YouTube. A subset of 10 k audio is transcribed. TED2 comes with 118 hours of English speech extracted from TED conference talks and its transcription for evaluating ASR. Its recordings are clear but with some reverberation. TED3 is an extension of TED2 and comprises 450 hours of talks. SWB is a 260-hour conversational speech recognition dataset containing two-sided telephone conversations. The TIMIT corpus was designed to provide read speech data and its word and phone-level transcriptions for acoustic-phonetic studies. It contains recordings in American English. Compared to the previous corpora labeled for ASR evaluation, VoxLingua107 consists of 6.6 k hours of audio in 107 languages and is annotated for language identification. Beyond the original purpose of evaluation, these corpora are also used in pre-training to improve the generalizability of learned representations.

For the purpose of pre-training and evaluating Mandarin speech representations, the authors of [101], [104] also compiled Open Mandarin, an open-source Mandarin dataset of 1.5 k hours of speech from the Linguistic Data Consortium (LDC) and OpenSLR.[2] Open Mandarin consists of the HKUST Mandarin Telephone Speech Corpus (HKUST, 200 hours of spontaneous speech, of which 168 hours of audio is used for pre-training; the development and test sets are excluded.) [225], AISHELL-1 [226] (178 hours of read speech), aidatatang 200zh (200 hours, read speech) [227], MAGICDATA Mandarin Chinese Read Speech Corpus (755 hours, read speech) [228], Free ST Chinese Mandarin Corpus (ST-CMDS, 100 hours, read

[1][Online]. Available: https://librivox.org/

[2][Online]. Available: https://openslr.org

speech) [229], and Primewords Chinese Corpus Set 1 (100 hours, read speech) [230]. Both HKUST and AISHELL-1 are labeled and are suitable for ASR evaluation.

C. Datasets for Evaluation

Besides the aforementioned datasets, conventional speech processing benchmarks are also used to evaluate self-supervised representations. Studies leverage Hub5, DIRHA, and CHiME-5 to measure the efficacy of representations in ASR. The Hub5 evaluation (LDC2002T43 and LDC2002S09, also referred to as the NIST 2000 Hub5 English evaluation set) contains 40 transcribed English telephone conversations only for testing, where 20 are from conversations collected in SWB studies but not released with the SWB dataset, and the rest are from CallHome American English Speech (LDC97S42). DIRHA [231], short for Distant-speech Interaction for Robust Home Applications, is a database composed of utterances sampled from WSJ, speech of keywords and commands, and phonetically-rich sentences. These utterances are read by U.K. and US English speakers and recorded with microphone arrays. CHiME-5 [232] is a challenge that aims to advance robust ASR and presents a dataset of natural conversational speech collected under a dinner party scenario with microphone arrays. A team at Amazon Alexa also recorded and transcribed a corpus of 1k hours of audio for model training and evaluation [126].

Researchers also evaluate representations for sentiment analysis with the INTERFACE [233] and MOSEI (CMU Multimodal Opinion Sentiment and Emotion Intensity) [234] datasets. INTERFACE is an emotional speech database for Slovenian, English, Spanish, and French, and contains six emotions: anger, sadness, joy, fear, disgust, and surprise, plus neutral. MOSEI is comprised of sentence-level sentiment annotations of 65 hours of YouTube videos using emotion categories similar to INTERFACE, but replacing joy with happiness.

In addition, datasets employed to demonstrate the benefit of SSL representations on various tasks include VCTK [235] and VoxCeleb1 [236] for SID/ASV (automatic speaker verification) tasks, FSC (Fluent Speech Commands) [237] for IC (intent classification), QUESST (QUESST 2014) [238] for QbE (query by example), LS En-Fr [239] and CoVoST-2 [240] for ST (speech translation), and ALFFA and OpenSLR-multi for multilingual ASR. The VCTK corpus includes speech data with 109 English speakers of various accents, each reading out about 400 sentences sampled from newspapers. VoxCeleb1 is an audio-visual dataset comprised of short YouTube clips containing human speech. It consists of 1251 unique speakers and 352 hours of audio. FSC contains utterances of spoken English commands that one might use for a smart home or virtual assistant, and is used to evaluate the performance of a spoken language understanding system. The QUESST search dataset comprises spoken documents and queries in 6 languages to measure the capability of models in spotting spoken keywords from documents. LS En-Fr is a dataset augmenting existing LS monolingual utterances with corresponding French translations to train and evaluate English-French machine translators. CoVoST-2 is a multilingual speech translation benchmark based on CV. It provides data for translating from English into 15 languages and from 21 languages into English, and has a total of 2.9 k hours of speech. The ALFFA project[3] collects speech of African languages to promote the development of speech technologies in Africa, and [123] leverages four African languages collected in the project for evaluation: Amharic [241], Fongbe [242], Swahili [243], and Wolof [244]. In the same work [123], the authors further select 21 phonetically diverse languages from OpenSLR to evaluate the generalizability of SSL representations across languages. We denote the collection as OpenSLR-multi below.

Last, [116] puts together five datasets (MUSAN [245], Bird Audio Detection [246], Speech Commands [247], Spoken Language Identification [248], and TUT Urban Acoustic Scenes 2018 [249]) plus an SID task built with the LS *train-clean-100* subset to evaluate the capability of representations on audio event detection. [117] employs the NSynth dataset [250] on top of the six for benchmarking. As many of the datasets are built for research in audio processing, we here provide only a list of these datasets for reference.

D. Experiment Settings for Evaluating SSL Techniques

A common way to benchmark SSL techniques and show their efficacy is to fine-tune a pre-trained SSL model for a supervised downstream task. Depending on the corpora used in pre-training and fine-tuning, techniques can be benchmarked in terms of their capability to transfer knowledge across datasets (i.e., using pre-training corpora that differ from the fine-tuning ones), their benefit when training with limited labeled examples (i.e., sampling a subset of labeled examples for fine-tuning), or their improvement over a fully supervised baseline (i.e., using the entire training split of downstream datasets for fine-tuning). Tables III and IV summarize experiment settings used in the SSL literature, including the pre-training corpora, downstream tasks and datasets, and the amount of fine-tuning labels used, which indicates the targeted benchmarking scenario as discussed above. Note that there are a variety of ways to fine-tune pre-trained networks (e.g., fine-tune the entire network, freeze certain layers during fine-tuning, and add various architectures of prediction layers to pre-trained networks). We here omit descriptions of these choices; readers can consult the original publications for details.

As observed in Tables III and IV, LS and WSJ are the most commonly used pre-training corpora. At the same time, we observe a growing industry investment in pre-training with larger datasets, e.g., CPC-8 k (8 k hours) for Bidir-CPC [123], LL (60 k hours) for CPC modified [122], wav2vec 2.0 [125], and HuBERT [130], Alexa internal datasets (10 k hours) for wav2vec-c [126], Didi internal datasets (10 k hours) for MPC [101], [104], the combination of Gigaspeech, VP-24 k, and LL (94 k hours in total) for UniSpeech-SAT [219] and WavLM [131],

[3][Online]. Available: http://alffa.imag.fr

TABLE III
A SUMMARY OF COMMON EXPERIMENT SETTINGS FOR VARIOUS SSL EVALUATIONS (PART 1). NETWORKS ARE USUALLY PRE-TRAINED WITH SSL TECHNIQUES, AUGMENTED WITH PREDICTION HEADS, AND FINE-TUNED (OR TRAINED) WITH LABELED DATA IN DOWNSTREAM TASKS FOR BENCHMARKING

Work	Pre-training corpus	Task	Dataset		Transfer	Fine-tuning labels used
			Training (fine-tuning)	Test		
CPC [56]	LS 100 hrs	PC	LS 100 hrs[14]	LS 100 hrs[14]	-	80[19] hrs
		SID	LS 100 hrs[14]	LS 100 hrs[14]	-	80[19] hrs
PASE [69]	LS 50 hrs [251]	SID	VCTK[15]	VCTK[15]	✓	44 hrs
		Sentiment	INTERFACE[16]	INTERFACE[16]	✓	3 hrs
		PR	TIMIT[15]	TIMIT[15]	✓	4 hrs
		ASR	DIRHA[15]	DIRHA[15]	✓	11 hrs
Audio2Vec [116]	AudioSet	AED	6 AED datasets[17]	6 AED datasets[17]	✓	See [116] for details
APC [92], [93]	LS 360 hrs	ASR	WSJ si284[21]	WSJ dev93	✓	72 hrs
		ST	LS En-Fr[15]	LS En-Fr[15]	-	236 hrs
		SID	WSJ si284[20]	WSJ si284[20]	✓	65[19] hrs
wav2vec [121]	LS 80/960 hrs, LS 960 hrs + WSJ si284	ASR	WSJ si284	WSJ eval92	✓	81 hrs
		PR	TIMIT[15]	TIMIT[15]	✓	4 hr
PhasePredict [117]	AudioSet	AED	7 AED datasets[17]	7 AED datasets[17]	✓	See [117] for details
Bidir-CPC [123]	LS 960 hrs, CPC-8k[18]	ASR	WSJ si284, LS 960 hrs, TED3[15]	WSJ eval92, LS test-clean, LS test-other, TED3[15], SWB[15]	Different datasets for training and test	81/960/450 hrs
		ASR-multi	ALFFA[15]	ALFFA[15]	✓	4 languages, 5.2–18.3 hrs
		ASR-multi	OpenSLR-multi[15]	OpenSLR-multi[15]	✓	21 languages, 4.4–265.9 hrs
MockingJay [100]	LS 360 hrs	PC	LS 360 hrs	LS test-clean	-	0.36[19]/1.8[19]/3.6[19]/18[19]/45[19]/360 hrs
		SID	LS 100 hrs[22]	LS 100 hrs[22]	-	90[19] hrs
		Sentiment	MOSEI[15]	MOSEI[15]	✓	65 hrs
CPC modified [122]	LS 100 hrs	PC	LS 100 hrs[14]	LS 100 hrs[14]	-	80[19] hrs
	LS 100 hrs, LS 960 hrs, LL 60k hrs	PC	CV-dataset[15]	CV-dataset[15]	✓	1 hrs
vq-wav2vec [124]	LS 960 hrs	ASR	WSJ si284	WSJ eval92	✓	81 hrs
		PR	TIMIT[15]	TIMIT[15]	✓	4 hrs
DeCoAR [99]	LS 100/360/460/ 960 hrs, WSJ si284	ASR	WSJ si284	WSJ eval92	-	25[19]/40[19]/81 hrs
		ASR	LS 100/360/ 460/960 hrs	LS test-clean, LS test-other	-	100/360/460/960 hrs
MT-APC [97]	LS 360 hrs	ASR	WSJ si284[21]	WSJ dev93	✓	72 hrs
		ST	LS En-Fr[15]	LS En-Fr[15]	-	236 hrs
PASE+ [113]	LS 50 hrs [251]	PR	TIMIT[15]	TIMIT[15]	✓	4 hrs
		ASR	DIRHA[15]	DIRHA[15]	✓	11 hrs
		ASR	CHiME-5[15]	CHiME-5[15]	✓	50 hrs
AALBERT [119]	LS 360 hrs	PC	LS 100 hrs[20]	LS 100 hrs[20]	-	80[19] hrs
		SID	LS 360 hrs[20]	LS 360 hrs[20]	-	288[19] hrs

The **Pre-training corpus, training (fine-tuning)**, and **test** columns list the datasets used in each work, and the **task** column lists the Tasks performed in the corresponding papers. We follow the abbreviation introduced in table II. The **transfer** column indicates whether The ssl technique is evaluated by its capability for transfer learning, i.E., Different datasets are utilized for pre-training and Fine-tuning. The **fine-tuning labels used** column summarizes the amount of labeled examples used in downstream fine-tuning.

[14]Train/test split made available by [56] on Google drive https://drive.google. com/drive/folders/1BhJ2umKH3whguxMwifaKtSra0TgAbtfb.

[15]Utilizes official training or test split.

[16]English utterances used in experiments. The utterances correspond to approximately 3 hours for training, 40 minutes for development, and 30 minutes for testing.

[17]The 6 AED datasets used in [116] are MUSAN [245], Bird Audio Detection [246], Speech Commands [247], Spoken Language Identification [248], TUT Urban Acoustic Scenes 2018 [249] plus an SID task built with LS train-clean-100. In addition to the 6 datasets, [117] use the NSynth dataset [250] for evaluation.

[18]A collection of AudioSet, AVSpeech, CV-dataset, LS,WSJ, TIMIT, Speech Accent Archive (SSA) [253], TED3, and SWB. SSA is a growing annotated corpus of English speech with various accents. Among the papers studied in this review, SSA is used in [123] only for pre-training, and only 1 h of audio is utilized. Thus, we exclude it from our discussion in Section IV.

[19]A subset of the official training split is sampled, usually to mimic lowresource learning conditions or to quickly evaluate for training and testing on the same split but disjoint subsets.

[20]Dataset split into training, validation, and test subsets at a ratio of 8:1:1.

[21]Dataset split into training and validation subsets at a ratio of 9:1.

[22]The dataset split into training and test subsets at a ratio of 9:1.

and the combination of VP-400 K, MLS, CV-dataset, VL and BBL (436 k hours in total) for XLS-R [218]. We expect this trend to continue with the growth in available computing power. Most studies focuses on learning representations for English, whereas Chinese [101], [104] and multilinguality [123], [218] are also gaining attention. Compared to pre-training, datasets used for fine-tuning are more diverse and cover downstream tasks as varied as ASR, PR, PC, SID, AED, Sentiment, ST, and LID. For benchmarking training scenarios covering full supervision as well as limited resources, the amount of labeled examples used for fine-tuning also varies from several minutes up to 1 k hours. Recent benchmarks such as SUPERB [44] that

TABLE IV
A SUMMARY OF COMMON EXPERIMENT SETTINGS FOR VARIOUS SSL EVALUATIONS (PART 2)

Work	Pre-training corpus	Task	Dataset		Transfer	Fine-tuning labels used
			Training (fine-tuning)	Test		
BMR [103]	WSJ si284, LS 960 hrs	ASR	WSJ si284	WSJ eval92	-	81 hrs
		PR	WSJ si84/si284	WSJ dev93	-	15/81 hrs
vq-APC [98]	LS 360 hrs	PC	WSJ si284[21]	WSJ dev93	✓	81 hrs
		SID	WSJ si284[20]	WSJ si284[20]	✓	65^{19} hrs
vq-wav2vec + DiscreteBERT [129]	LS 960 hrs	ASR	LS 100 hrs	LS test-clean, LS test-other	-	10 mins[19], $1^{19}/10^{19}$/100 hrs
speech-XLNet [109]	LS 960 hrs + WSJ si284 + TED2	PR	TIMIT[15]	TIMIT[15]	✓	4 hrs
		ASR	WSJ si284	WSJ eval92	-	$7^{19}/14^{19}/30^{19}$/81 hrs
MPC [101], [104]	Didi Callcenter, Didi Dictation, Open Mandarin	ASR-zh	HKUST[15]	HKUST[15]	✓	168 hrs
		ASR-zh	AISHELL-1[15]	AISHELL-1[15]	✓	178 hrs
	SWB, Fisher 1k, LS 960 hrs	ASR	SWB	Hub5'00	-	260 hrs
MPE [102]	WSJ si284, LS 960 hrs	ASR	WSJ si284	WSJ eval92	-	$25^{19}/40^{19}$/81 hrs
		ASR	LS 100/360/960 hrs	LS test-clean	-	100/360/960 hrs
ConvDMM [252]	LS 50 [251]/360/ 960 hrs	PC/PR	WSJ si284	WSJ eval92	✓	$5^{19}/50^{19}/100^{19}$ mins, $4^{19}/8^{19}/40^{19}$ hrs
wav2vec 2.0 [125]	LS 960 hrs, LL 60k hrs	ASR	LS 960 hrs	LS test-clean, LS test-other	-	10 mins[19], $1^{19}/10^{19}$/100/960 hrs
		PR	TIMIT[15]	TIMIT[15]	✓	4 hrs
NPC [107]	LS 360 hrs	PC	WSJ si284[21]	WSJ dev93	✓	81 hrs
		SID	WSJ si284[20]	WSJ si284[20]	✓	65^{19} hrs
DeCoAR 2.0 [110]	LS 960 hrs	ASR	LS 100 hrs	LS test-clean, LS test-other	-	$1^{19}/10^{19}$/100 hrs
TERA [106]	LS 100/360/ 960 hrs	PC	LS 100 hrs[14]	LS 100 hrs[14]	-	80^{19} hrs
		SID	LS 100 hrs[14]	LS 100 hrs[14]	-	80^{19} hrs
		PR	TIMIT[15]	TIMIT[15]	✓	4 hrs
		ASR	LS 100 hrs	LS test-clean	-	100 hrs
HuBERT [130]	LS 960 hrs, LL 60k hrs	ASR	LS 960 hrs	LS test-clean LS test-other	-	10 mins[19], $1^{19}/10^{19}$/100/960 hrs
wav2vec-c [126]	Alexa-10k	ASR	Alexa-eval	Alexa-eval	✓	1k hrs
UniSpeech-SAT [219]	LL 60k hrs + GigaSpeech-10k + VP-24k	Multi	SUPERB	SUPERB	✓	See SUPERB [44] paper for details
WavLM [131]	LL 60k hrs + GigaSpeech-10k + VP-24k	Multi	SUPERB	SUPERB	✓	See [44] for details
XLS-R [218]	VP-400k + MLS + CV-dataset-7k + VL + BBL	ASR	VP, MLS, CV-dataset, BBL, LS	VP, MLS, CV-dataset, BBL, LS	-	See [218] for details
		SID	VoxCeleb1	VoxCeleb1	✓	
		ST	CoVoST-2	CoVoST-2	✓	
		LID	VL	VL	-	

See the caption of table iii for a detailed description of all the abbreviations used in this table.

consolidate multiple downstream tasks have gained attention for evaluating SSL methodologies [131], [219]. The goal of such benchmarks is to provide a holistic evaluation of the performance of learned representations; we discuss these in detail in Section IV-E. With the increasing popularity of SSL research, we expect future experiment settings to proliferate and cover more languages, downstream tasks, and pre-training/fine-tuning datasets.

E. Benchmark Results and Discussion

Given the diversity of datasets and downstream tasks used to evaluate SSL techniques in the literature, it is infeasible to discuss all experiment settings in this survey. Hence, due to their wide adoption for experiments conducted by studies in both SSL and the speech community in general, we focus first on ASR on the LS dataset to understand the efficacy of SSL. We examine SSL techniques which report ASR results on the LS *test-clean* split, and summarize the published WER in Fig. 3. The ASR models were obtained first by using unlabeled speech to pre-train a model with each SSL technique. The model was then fine-tuned on labeled data by utilizing a supervised training objective. Respectively, 960, 100, 10, 1 h(s), and 10 minutes of labeled LS training data were used for fine-tuning, as indicated in different panels of Fig. 3 (see the caption of Fig. 3 for more details). Semi-supervised methods such as self-training, where a model is first trained on labeled data to annotate unlabeled speech, and then subsequently trained on combined golden and self-annotated label-speech pairs, are gaining popularity in the speech community and have yielded competitive results. For comparison, we also show performance from such methods (iterative pseudo labeling (IPL) [63], slimIPL [254], noisy student [62]), as well as the current state of the art—conformer XXL + noisy student [255]—which augments SSL with various advanced techniques including self-training. Furthermore, we illustrate in the figure the performance of a baseline system [44] based on log mel filterbank (fBANK), which is one of the most commonly used features designed by domain experts. As observed in the figure, most SSL techniques outperform fBANK features, and with the growing investment in model size, better performance is achieved. The largest ones, such as wav2vec 2.0-L and HuBERT-L/XL, yield competitive results when the

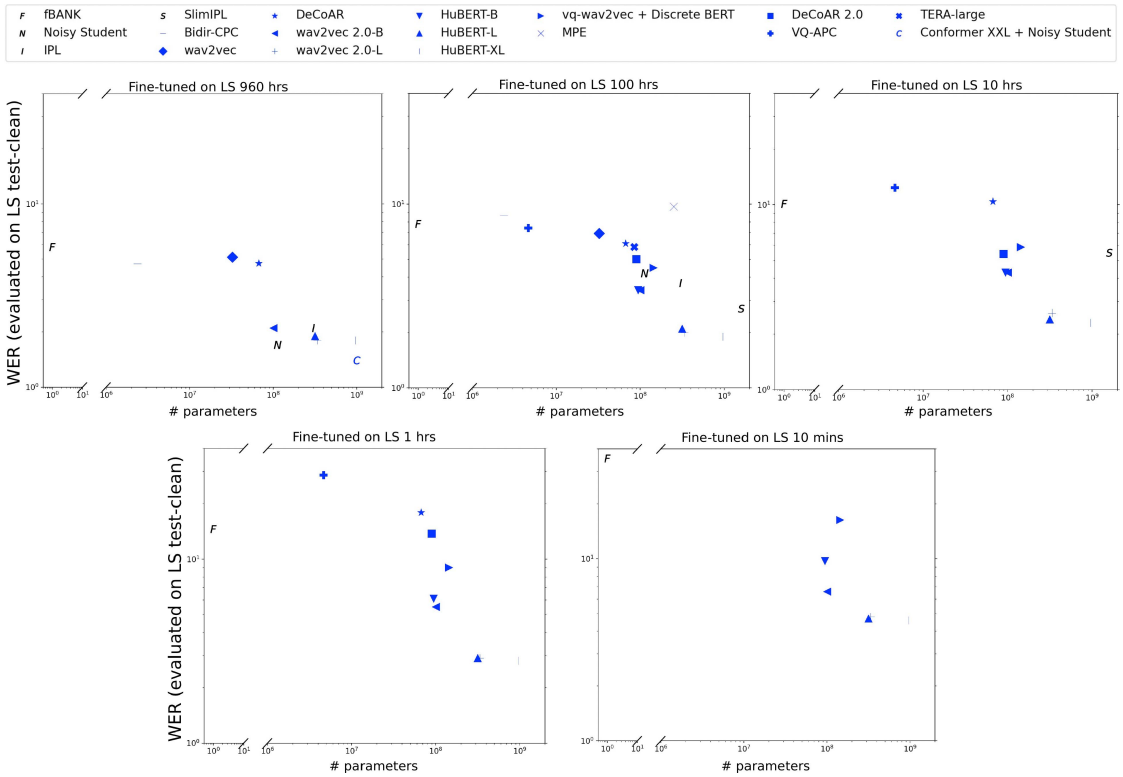

Fig. 3. SSL performance on ASR WER (vertical axis) evaluated with LS *test-clean* split. Techniques are sorted based on the number of model parameters along the horizontal axis. Markers in blue correspond to models initialized with various SSL techniques and then fine-tuned using 960, 100, 10, 1 h(s), and 10 minutes respectively. The 960-hour training set is the aggregation of *train-clean-100*, *train-clean-360*, and *train-other-500* splits. The 100-, 10-, 1-hour, and 10-minute sets leverage *train-clean-100* or its sampling, except for Bidir-CPC, which samples 10% of the training examples from the entire 960-hour corpus. For simplicity, several SSL techniques are appended with suffixes *B*, *L*, *XL*, or *XXL* indicating the *Base*, *Large*, *X-Large*, or *XX-Large* variants specified in the original publication. We also compare with baselines including the log mel filterbank (fBANK) and semi-supervised, self-training approaches (iterative pseudo labeling (IPL) [63], slimIPL [254], noisy student [62]). These approaches are visualized in black. Also, note that the current state of the art—conformer XXL + noisy student [255]—is a combination of self-training and SSL techniques. Given the diversity of the listed methods in experiment settings (e.g., pre-training corpora and objectives, whether a language model is used in decoding, whether model parameters are frozen in fine-tuning), readers should be careful that the superiority of methods cannot be decided only based on lower WER numbers.

entire 960-hour of labeled data is used in training/fine-tuning. The benefit of SSL, especially models with more parameters like wav2vec 2.0 and HuBERT, becomes more evident when the labeling resources become scarce. Compared to popular semi-supervised methods such as IPL, slimIPL, and noisy student using 100 hours of labels, wav2vec 2.0 and HuBERT achieve lower or competitive WERs with 1 h or even 10 minutes of labeled examples. The results are highly favorable for low-resource use cases, for instance when expanding systems to new domains or languages for which large amounts of unlabeled audio are available, since collecting labels for new conditions is often prohibitively slow or costly.

In addition to the ASR task, where the current state of the art is achieved by a method combining SSL pre-training and self-training techniques [255], SSL models are competitive in other tasks, including IC, SID, ASV, and QbE. We summarize the performance of these models and previous non-SSL methods in Table V. The results suggest that the benefit of SSL is generalizable among tasks that require encoding information such as content, speaker, and semantics. As SSL research gains more attention, we expect that SSL pre-trained models will achieve state-of-the-art results on an increasing number of tasks.

TABLE V
TASKS WHERE THE STATE OF THE ART IS MODELS WITH SSL PRE-TRAINING

Tasks	Dataset	non-SSL	SSL
ASR (WER ↓)	LS test-clean/other	2.1/4.0 [63]	1.4/2.6 [255]
IC (Acc ↑)	FSC	98.8 [237]	99.3 [219]
SID (Acc ↑)	VoxCeleb1	94.8 [256]	95.5 [131]
ASV (EER ↓)	VoxCeleb1	3.1 [257]	2.4 [258]
QbE (MTWV ↑)	QUESST (EN)	10.6 [259]	11.2 [219]

Despite the obvious trend of increasing performance as more parameters and SSL pre-training data are being used, numbers in Fig. 3 and Table V are less comparable than might be expected. The task performance is obtained from the original papers and is often achieved with different downstream fine-tuning recipes, including various language models (used in the ASR system), prediction heads (networks added to SSL for downstream inference), or choices between fine-tuning the whole networks or freezing the SSL encoders. For example, in the ASR task, HuBERT-L and wav2vec 2.0-L leverage Transformer as their language model, while a 4-gram language model trained on LS is used in DeCoAR 2.0. The lack of common and

established mechanisms to evaluate SSL techniques in downstream applications makes it difficult to compare techniques fairly and understand their capabilities. To address this challenge, there are increasing efforts to establish benchmarks with shared downstream tasks, datasets, and downstream recipes. Such efforts include SUPERB [44], LeBenchmark [260], ZeroSpeech [261], HEAR [262], NOSS [263], and HARES [264].

SUPERB [44] is a benchmarking platform that allows the SSL community to train, evaluate, and compare speech representations on diverse downstream speech processing tasks, from acoustic and speaker identity to paralinguistic and semantics. SUPERB consolidates downstream recipes to focus on common and straightforward settings (e.g., prediction head architectures, language models, hyperparameter spaces) to facilitate generalizable and reproducible benchmarking of SSL techniques. SUPERB also encourages researchers to innovate for efficient use of model parameters and computation resources to democratize SSL beyond race among Big Tech. LeBenchmark [260] shares a vision similar to SUPERB and provides a reproducible framework for assessing SSL in French with ASR, spoken language understanding, speech translation, and emotion recognition. ZeroSpeech [261] (described in more detail in Section VI-B1) challenges the scientific community to build speech and language understanding systems using zero expert resources for millions of users of "low-resource" languages. SSL techniques are also benchmarked with the ZeroSpeech challenge [265], [266]. Apart from the speech community, researchers have also established HEAR (holistic evaluation of audio representations) [262], NOSS (non-semantic speech benchmark) [263], and HARES (holistic audio representation evaluation suite) [264] to benchmark audio representations. These efforts promote the creation of an audio embedding that is as holistic as the human ear in interpreting speech, environmental sound, and music. Given the significant need to understand and compare SSL techniques fairly and comprehensively, we expect SSL benchmarking to remain an active research area.

V. Analysis of Self-Supervised Representations

The previous sections have shown how self-supervised learning can result in powerful representations that provide a robust starting point for several downstream tasks. It is natural to ask if we can gain an even deeper understanding of the nature of these representations, in order to further optimize them or apply them to different problems. What is the information encoded in these representations? How robust are they to distributional shifts, and how dependent are they on the size of the training data? Do they generalize across languages? What are the key ingredients for training powerful representations: input data, network architecture, training criterion, or all three? Can we predict their performance on downstream tasks from their training behavior? This section tries to answer these questions by summarizing several studies that analyze self-supervised representations.

A. Information Content

In [68] wav2vec 2.0 representations were analyzed with respect to their acoustic-linguistic information content at different network layers. Three different mechanisms were used for this purpose. The first of these is canonical correlation analysis (CCA), which computes similarity scores between two continuous vectors based on the maximum correlation of their linear projections. These can be used to judge the similarity of embeddings at different layers with each other, with standard acoustic representations such as mel filterbank features, or word embeddings derived from text. The second method clusters continuous representation vectors and computes the discrete mutual information between cluster IDs and phone or word labels. The third method involves probing tasks: representation vectors extracted from the network are used to perform simple downstream tasks, in particular determining whether two acoustic segments correspond to the same word, and a standard benchmark of 11 word similarity tasks [267]. These are mostly used to gauge the amount of lexical information present in the embeddings. Using this battery of tests the authors compared pre-trained models of varying sizes as well as models fine-tuned for ASR. They found that pre-trained models show an autoencoder-style behavior, with early layers showing strong similarity with input features, intermediate layers diverging more, and final layers reverting back to higher similarity with input features and early layers. Generally, the earlier layers in wav2vec 2.0 models encode acoustic information. The next set of layers encodes phonetic class information, followed by word meaning information, before reverting back to encoding phonetic/acoustic information. Thus, extracting representations from the last layers for tasks that require phonetic or word-related information may not be the best strategy. Indeed, the authors of [268] show that a phone classifier trained on each of the 24 frozen layers of a wav2vec 2.0 model showed the lowest phone error rates for layers 10–21 and higher error rates for the other layers. [68] further show that fine-tuning the pre-trained model with a character-level CTC training criterion changes the behavior of the last layers (especially the final two layers), breaking the autoencoder-style behavior and focusing the information encoded in the last layers on orthographic-phonetic and word information.

The peaking of class-relevant information in intermediate layers seems to be common across different self-supervised learners and different modalities. In an analysis of text-based Transformers trained with a masked language model criterion [269] observed a similar compression plus reconstruction pattern. Interestingly, similar network behavior was also recently described for self-supervised learners in computer vision: using a contrastive self-supervised learner (SimCLR) that optimizes for augmentation invariance, [270] show that it is the intermediate representations that most closely approximate information learned in a supervised way, i.e., they provide more class information than the representations from final layers. This is similar to the findings described above for wav2vec 2.0 without fine-tuning, where intermediate layers provide more information about phone and word classes.

Self-supervised representations may encode other information besides phonetic classes or words, for example, channel, language, speaker, and sentiment information. It is shown that the per-utterance mean of CPC features captures speaker information to a large extent [271]. Location of information pertaining to speakers vs. language classes was analyzed in [272] for a 12-layer BERTphone model. This model combines a self-supervised masked reconstruction loss with a phone-based CTC loss to produce representations for speaker recognition and language identification. By analyzing the weights of a linear combination of layer representations for these two downstream tasks, it was shown that language recognition draws on representation from higher layers (peaking at layer 10) whereas speaker recognition benefited from layers at positions 6, 9, and 12. This may indicate that language recognition relies more on higher-level phonetic information whereas speaker recognition uses a combination of acoustic and phonetic information. In a recent study [131] the same technique was used to identify layer contributions for the downstream SUPERB benchmark tasks in the WavLM model. For a smaller model (95 M parameters) it was again confirmed that lower layers encode speaker-related information necessary for speaker diarization and verification whereas higher layers encode phonetic and semantic information. Another study [273] used explicit self-supervised loss at the intermediate layers rather than just the output layer of a HuBERT model in order to enforce better learning of phonetic information. The resulting model was indeed better at downstream tasks requiring information about phonetic content, such as phone recognition, ASR, and keyword spotting, but worse at speaker-related tasks like speaker diarization and verification.

Most self-supervised learning approaches rely on a Transformer architecture for the representation model. In [274] the attention patterns in generatively trained Transformer representation models were analyzed. Self-attention heads were grouped into three categories: diagonal, vertical, and global. It was found that the diagonal head focuses on neighbors and is highly correlated with phoneme boundaries, whereas the vertical head focuses on specific phonemes in the utterance. Global heads were found to be redundant as removing them resulted in faster inference time and higher performance.

B. Training Criterion

In [275], representations based on different training criteria (masked predictive coding, contrastive predictive coding, and autoregressive predictive coding) were compared and analyzed with respect to the correlation between their training loss and performance on both phone discrimination and speaker classification probing tasks. It was observed that the autoregressive predictive coding loss showed the strongest correlation with downstream performance on both tasks; however, models were not further analyzed internally. An evaluation of the similarity of representations trained according to the three criteria above (but with different architectures and directionality of contextual information) also showed that it is the training criterion that most influences the information encoded in the representations, not the architecture of the learner or the directionality of the input.

A similar insight was obtained in [276], which compared vq-vae and vq-wav2vec with respect to their ability to discover phonetic units. The vq-vae model extracts continuous features from the audio signal; a quantizer then maps them into a discrete space, and a decoder is trained to reconstruct the original audio conditioned on the latent discrete representation and the past acoustic observations. By contrast, vq-wav2vec predicts future latent discrete representations based on contextualized embeddings of past discrete representations, in a CPC-style way. The models were evaluated according to their ability to discover phonetic units (as measured by phone recognition error rate on TIMIT, and the ZeroSpeech ABX task (see Section VI-B for more details)), and it was found that the predictive vq-wav2vec model fared better than the autoencoder-like vq-vae model, most likely due to its superior ability to model temporal dynamics.

C. Effects of Data and Model Size

How does the performance of self-supervised models change in relation to the amount of training data, and in relation to the size (number of parameters) of the model? Several studies have demonstrated better downstream performance when using larger datasets [123], [131], [277]. For example, [123] compared representations learned by a bidirectional CPC model from the standard 960 h LS corpus and a corpus of 8,000 hours of diverse speech from multiple sources.[19] Not surprisingly, an ASR model trained on top of these representations performed better when representations were learned from the larger dataset. Although the precise relationship between data size and performance has not been quantified, we can assume that it follows a law of diminishing returns (or power law), similar to observations for most data-intensive machine learning tasks. In addition to the size of the dataset, the diversity of the data also seems to play a role, although this was not quantified in this study. However, recent experiments with larger and more diverse data collections [131] confirm this assumption, as do explicit investigations of domain shift robustness (see Section V-D below).

The relation between model sizes and downstream performances have also been investigated [278], [279]. Using the Mockingjay model [100], the authors in [278] attempt to establish a relationship between model size and self-supervised L_1 loss and demonstrate that it approximately follows a power law. Model size and accuracy on downstream phone classification and speaker recognition tasks are positively correlated but do not exactly follow a power law; rather, the accuracy saturates as models increase in size, possibly due to the lack of a corresponding expansion in training data size.

D. Robustness and Transferability

It is well known that traditional speech features like MFCCs lack robustness against environmental effects such as additive noise, reverberation, accents, etc., that cause differences in the distributions of speech features. Do pre-trained representations offer greater robustness against distributional shifts? One study [123] compared pre-trained representations from a CPC model against MFCCs and found pre-trained representations to be more robust to mismatches between training and test data.

The training data consisted of clean, read speech (LS) whereas test data consisted of the Switchboard corpus and TED talks. The distributional shifts here may stem from both the acoustics (microphone, room reverberation) as well as lexical effects related to topic and style, as well as differences in speaker characteristics such as accent. Similar problems were also investigated using HuBERT and wav2vec 2.0 models in [280]. In [281] domain effects were studied in greater detail using datasets from six different domains. In particular, the authors focused on the usefulness of adding out-of-domain data to pre-training. The general conclusions are that pre-training on more and diverse domains is preferable: models pre-trained on more domains performed better than those pre-trained on fewer when tested on held-out domains, regardless of which additional labeled data was used for fine-tuning. Adding in-domain unlabeled data—if available—to pre-training improves performance robustly; however, even out-of-domain unlabeled data is helpful and closes 66–73% of the performance gap between the ideal setting of in-domain labeled data and a competitive supervised out-of-domain model.

In [277] the effectiveness of CPC-trained representations for phone discrimination tasks was compared across several languages. It was found that representations pre-trained only on English successfully enabled phone discrimination in 10 other languages, rivaling supervised methods in accuracy in low-data regimes (1 h of labeled data per language). Thus, self-supervised pre-training enables the model to learn contextualized speech features that generalize across different languages. In [282], a wav2vec 2.0 model was trained on data from multiple different languages and different corpora (Babel, Common Voice, and multilingual LS) jointly, followed by fine-tuning for each individual language. The largest model covers 53 languages in total and consists of 56,000 hours of speech. Compared to monolingual pre-training, even smaller models trained on only ten languages improve performance substantially on a downstream character-based ASR task. Low-resource languages with little labeled data improve the most under this training regime. Multilingual representations also resulted in competitive performance (lower character error rate than monolingual representations) for languages not present in the training dataset, again showing that unsupervised pre-trained representations can learn generic features of the speech signal that generalize across different languages. The study also found that sharing data from closely related languages is more beneficial than combining distant languages. An analysis of language clusters in the shared discrete latent representation space revealed that similar languages do indeed show a higher degree of sharing of discrete tokens. Finally, one might ask whether the interpretation of representations extracted from different layers of a self-supervised models also generalizes to the multilingual setting. Experiments in [268] on phone recognition in eight languages based on the different layers of the multilingual wav2vec 2.0 XLSR-53 model indicate that this is indeed the case: phone error rates showed the same pattern as in the monolingual (English) scenario, with lower phone error rates for middle layers as opposed to earlier/later layers.

VI. From Representation Learning to Zero Resources

In the SSL framework, speech representations can be learned and used in various downstream tasks to achieve competitive, robust, and transferable performance, as shown in Section IV to V. However, labeled data is still required. For example, in ASR, utterances and their manual transcriptions are needed to learn downstream models or fine-tune representation models. Can a model learn without any labeled data? In Section VI-A, we show how to learn ASR models without any paired audio and text and how SSL improves the framework. In addition, many languages have no writing system. In Section VI-B, the SSL representation is further used in scenarios where text data is unavailable.

A. Unpaired Text and Audio

1) Unsupervised ASR: If only unpaired speech and text are available, that is, the text is not a manual transcription of speech, can the machine learn how to transcribe speech into text? This scenario is called *unsupervised ASR*, and the framework is as below. Given a set of unlabeled utterances $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$ and a set of sentences $\mathcal{Y} = \{Y_1, Y_2, \ldots, Y_M\}$,[4] a mapping function F, which can take an utterance S as input and generate its transcription, is learned from data. Table VI summarizes recent work on unsupervised ASR, including the speech representation used, the algorithm used to learn the mapping without supervision, and the results. Below, we will discuss these methods in more detail.

Adversarial training [285], [297], [298] is one common way to learn such a mapping function. The framework includes a discriminator and a generator. The mapping function F plays the role of the generator, which takes speech utterances as input and outputs text. The discriminator learns to distinguish real text from the generated output; the generator learns to "fool" the discriminator. The generator and the discriminator are trained in an iterative, interleaved way. After the training, the generator serves as the speech recognition model. There is a large amount of work using gradient penalty in the objective of training discriminators [268], [283], [293], [296], which is inspired by Improved Wasserstein Generative Adversarial Network (WGAN) [285]. Other ways to map speech and text include via segmental empirical output distribution matching (segmental empirical-ODM) [290] and decipherment algorithm [294].

Success in unsupervised neural machine translation (MT) [287], [299], [300] has inspired innovative exploration of various unsupervised ASR algorithms. If learning a translation model from unaligned sentences in two languages is possible, considering speech and text as two different languages, learning the mapping relationship from speech space to text space without an alignment should likewise be possible. However, there are differences between unsupervised MT and unsupervised ASR. In unsupervised MT, most discrete source tokens can be mapped to specific target tokens representing the same meaning. However, because speech has segmental structures,

[4]Note that the speech and text are not paired, that is, Y_i is not the transcription of S_i.

TABLE VI
UNSUPERVISED ASR. TIMIT NUMBERS ARE PHONEME ERROR RATES (PER), WHILE THE NUMBERS FOR LIBRISPEECH ARE WORD ERROR RATES (WER)

Reference	Speech representation	Speech segmentation	Text token/representation	Mapping approach	Refinement	Results
[283]	Audio word2vec [284]	Oracle	Phoneme	Adversarial Training [285]	-	TIMIT (PER): 63.6%
[286]	Speech2vec [114]	BES-GMM [196]	Word2Vec	Adversarial Training [287]	Self-training	SWC (Acc): 10.9%
[288]	Speech2vec (English)	Oracle	Word2Vec (French)	VecMap [289]	LM rescore, sequence DAE	ST (BLEU): 10.8%
[290]	MFCC	GAS [291]	Phoneme	Empirical-ODM [292]	Self-training	TIMIT (PER): 41.6%
[293]	MFCC	GAS	Phoneme	Adversarial Training [285]	Self-training	TIMIT (PER): 33.1%
[268]	Wav2vec 2.0 [125]	k-means	Phoneme	Adversarial Training [285]	Self-training	TIMIT (PER): 18.6%, LibriSpeech (WER): 5.9%
[294]	Universal Phone Recogniser	-	Grapheme	Decipherment [295]	Self-training	GlobalPhone: 32.5% to just 1.9% worse than supervised models
[296]	Wav2vec 2.0 [125]	-	Phoneme	Adversarial Training [285]	Self-training	LibriSpeech (WER): 6.3%
[296]	Wav2vec 2.0 [125]	-	Grapheme	Adversarial Training [285]	Self-training	LJSpeech (WER): 64.0%

SWC = spoken word classifier, ST = speech translation. All speech and text are in english if not specified. The references in the table are sorted according to the date of publication.

in unsupervised ASR, each text token maps to a segment of consecutive acoustic features of variable length in an utterance. The generator is supposed to learn the segmental structure of an utterance because information like token boundaries is not directly available. This makes unsupervised ASR more challenging than unsupervised MT.

For unsupervised ASR to be feasible, the common idea is to make the speech and text units close to each other. For the text side, word sequences can be transformed into phoneme sequences if a lexicon is available. On the other hand, we must first convert the speech signal into something close to phonemes. To achieve that, most studies on unsupervised ASR use a phoneme segmentation module before the generator to segment utterances into phoneme-level segments [283], [290], [293]. A representation vector or a token then represents each phoneme-level segment. It is easier for the generator to map each segment-level representation or token to the correct phoneme when the representation or token is highly correlated to the phonemes. Wav2vec-U [268] selects the input feature from different layers of wave2vec 2.0 [125]. The selection criterion is based on analysis of the phonetic information in each layer. If a universal phone recognizer trained from a diverse set of languages is available, it is another way to transcribe speech into phone-level tokens [294]. Another series of work is to transform a word into a word embedding. [286], [288] use adversarial training to map the word-level speech embedding space [114] to the word embedding space and achieve promising performance on spoken word classification, speech translation, and spoken word retrieval. Table VI summarizes the various ways to segment speech and represent speech and text in each reference.

As shown in Table VI, most studies use *self-training* to refine the models. In self-training, the generator serves as the first-version phoneme recognition model. Inputting unpaired speech to the generator generates the corresponding "pseudo transcription". We then view the speech utterances and their pseudo transcriptions as paired data which we use to train a model in a supervised manner. Although the pseudo transcriptions have more errors than oracle transcriptions, experiments show that training models on pseudo transcriptions still significantly boosts performance compared to the first-version' model.

Wav2vec-U [268] achieved state-of-the-art results at the time, which suggests that representation learning is essential for the success of unsupervised ASR. It achieved an 11.3% phoneme error rate on the TIMIT benchmark. On the LS benchmark, wav2vec-U achieved a 5.9% WER on *test-other*, rivaling some of the best published systems trained on 960 hours of labeled data from only two years earlier. And wav2vec-U 2.0 [296] further removes the requirement of the segmentation stage, so the unsupervised ASR model can be learned in an end-to-end style. The robustness of wav2vec-U was further analyzed with respect to domain-mismatch scenarios in which the domains of unpaired speech and text were different [301]. Experimental results showed that domain mismatch leads to inferior performance, but a representation model pre-trained on the targeted speech domain extracts better representations and reduces this drop in performance.

2) Asr-Tts: Here we describe an alternative approach by which to train an ASR and text-to-speech (TTS) system based on unpaired text and audio. The ASR-TTS framework, which combines the ASR and TTS systems in a cascaded manner, can be regarded as an autoencoder, where the encoder f corresponds to the ASR module and the decoder g corresponds to the TTS module. In this framework, we consider the intermediate ASR output as a latent representation; the framework as a whole can be regarded as a variant of self-supervised learning.[5]

The ASR-TTS framework can jointly optimize both ASR and TTS without using paired data [302], [303], [304]. A speech chain [302], [305] is one successful way to utilize audio-only and text-only data to train both end-to-end ASR/TTS models. This approach first prepares pre-trained ASR model $f_{\text{asr}}(X)$ with acoustic input X and pre-trained TTS model $g_{\text{tts}}(Y)$ with text input Y. By following the TTS system with an ASR system, we generate new acoustic feature sequence $\hat{X}$, which must be close to the original input X. Thus, we design a loss function $\mathcal{L}_{\text{asr}\to\text{tts}}(X, \hat{X})$, where $\hat{X}$ is generated by

$$\hat{X} = g_{\text{tts}}(f_{\text{asr}}(X)). \tag{36}$$

[5]However, to make this complicated system work, we often require that data is paired. Therefore, in practice, ASR-TTS and other methods described in this section are categorized as semi-supervised learning.

Thus, we train the ASR model (or both ASR and TTS models) using only the acoustic input by minimizing $\mathcal{L}_{\text{asr}\to\text{tts}}$. Note that this approach does not require the supervised text data Y. As an analogy to the generative approach in Section III-B, the intermediate ASR output $\hat{Y}$ can be regarded as the latent representation Z.

The other cycle with the text-only data Y is also accomplished by the concatenated TTS-ASR systems:

$$\hat{Y} = f_{\text{asr}}(g_{\text{tts}}(Y)). \tag{37}$$

Similarly, this approach does not require the supervised audio data X, and the intermediate TTS output $\hat{X}$ can be regarded as the latent representation Z. Although this approach initially freezes either the ASR or TTS model, extensions of this study [303], [306], [307] implement the joint training of both ASR and TTS parameters using REINFORCE [308] and straight-through estimators.

An emerging technique uses a well-trained TTS system to generate speech and text data from text-only data. This technique is a sub-problem of the TTS-ASR approach formulated in (37) in which we fix the TTS system part and estimate only the ASR parameters. For example, a huge amount of text resources can be obtained from the web and document archives without corresponding audio data. The typical use case scenario of such a text resource for ASR is through the language model. We combine the ASR and language model via a noisy channel model [309], a weighted finite state transducer [310], or shallow fusion [311], [312]. However, the progress of TTS systems boosted by deep learning [80], [313] has inspired another interesting and straightforward research direction: artificially creating paired text and audio data $\{\hat{X}, Y\}$ with only text data Y by generating the corresponding audio data $\hat{X}$ with TTS. The most straightforward approach is to simply use multi-speaker TTS to generate the waveform with various acoustic variations [314], [315], [316], [317], [318]. The other approaches are based on the generation of high-level (more linguistic) features instead of generating the waveform, e.g., encoder features [319] and phoneme features [320], [321]. This approach is similar to the back-translation technique developed in neural machine translation [322]. One benefit of the above data generation approaches is that it can be used to feed unseen word or context phrases to end-to-end ASR.

B. No Text or Lexicon

1) Zero-Resource Speech Technologies and Challenges: Zero-resource speech technologies, which seek to discover linguistic concepts from audio only (no text nor lexicon), are one of the most active applications of unsupervised/self-supervised speech processing. Zero-resource speech technologies were initially studied for acoustic and linguistic unit discovery from speech data without linguistic resources, e.g., transcriptions and other annotations [323]. This study was motivated by unsupervised query-by-example, applications of non-parametric Bayesian machine learning to speech processing, and low-resource speech recognition, and was also inspired by the learning process of infants. The goal of this type of work is to build spoken dialog systems in a zero-resource setup for any language. To encourage zero-resource research, zero-resource speech challenges have been organized since 2015.

In this section, we describe the research directions of zero-resource speech technologies by following the series of zero-resource speech challenges.

- Zero Resource Speech Challenge 2015 [279] mainly focused on building an acoustic model without using any linguistic annotations based on subword unit modeling and spoken term discovery tracks. For the subword unit modeling track, the ABX score for the within- and across-speaker tasks was used as an evaluation metric. The spoken term discovery track used the normalized edit distance and coverage scores in addition to the precision, recall, and F1 scores for types, tokens, and boundaries. Both tracks were based on the English and Xitsonga languages.
- The Zero Resource Speech Challenge 2017 [324] focused on unseen language and speaker aspects from the previous challenge. For example, to demonstrate the robustness against unseen languages, the systems were developed with English, French, and Mandarin and tested on two "surprise" languages: German and Wolof. Similarly, robustness against unseen speakers was demonstrated by varying the amount of speech available for each speaker.
- The Zero Resource Speech Challenge 2019 [325] extended a goal of previous challenges by synthesizing speech without text or phonetic labels but with acoustic units obtained using zero-resource techniques. The evaluation metrics were also extended to subjectively evaluate the quality of synthesized speech, including its intelligibility, naturalness, and speaker similarity.
- The Zero Resource Speech Challenge 2020 [261] was based on two tracks, revisiting previous challenges with different evaluation metrics. The first task revisited the 2019 challenge with low bit-rate subword representations that optimize the quality of speech synthesis. The second task revisited the 2017 challenge by focusing on the discovery of word-like units from unsegmented raw speech.
- The Zero Resource Speech Challenge 2021 [326], the latest challenge, expanded the scope to include language modeling tasks. In addition to phoneme-level ABX, the challenge includes lexical, semantic, and syntactic evaluation metrics computed via a language model of pseudo-acoustic labels.

These challenges have facilitated the tracking of technical trends in zero-resource speech technologies. For example, research directions thereof have expanded to various speech processing components to cover the entire spoken dialogue systems. To keep up with this expansion, the challenge has continued to develop appropriate evaluation metrics for zero-resource scenarios. Following the success of representation learning, baseline and challenge techniques have shifted from purely generative models [327], [328], deep autoencoders [83], [329], and incorporation of neural-network-based TTS/VC techniques [266] to self-supervised learning [330]. The latest challenge included the visual modality, continuing the expansion to include more aspects of human interaction.

2) Textless NLP: Textless NLP is a new research direction that leverages the progress mentioned above in self-supervised speech representation learning to model language directly from audio, bypassing the need for text or labels [76], [77], [78], [331]. Not only does this open the gate for language and dialect modeling without orthographic rules, but it also offers the opportunity to model other non-lexical information about how speech is delivered, e.g., speaker identity, emotion, hesitation, interruptions. The generative spoken language model (GSLM) [78] utilizes discrete representations from wav2vec 2.0, HuBERT, and CPC algorithms as inputs to an autoregressive language model trained by using the cross-entropy function to maximize the probability of predicting the next discrete speech token. A synthesis module follows the language model to produce speech waveforms given the generated discrete speech units. The generated spoken continuations compete with supervised generations and synthesis using a character language model in subjective human evaluations. The model completes incomplete words (pow[..] $\rightarrow$ POWER) and continues using words in the same general mood (dark $\rightarrow$ BLACKNESS)[6] and has been extended to model and generate dialogues [332].[7] Given its flexibility in modeling spoken content, the GSLM has been further extended to jointly model content and prosody [77]. This prosodic-GSLM model introduced a multistream causal Transformer, where the input and output layers use multiple heads to model three channels: discrete speech units, duration, and quantized pitch. The prosodic-GSLM model jointly generates novel content and prosody congruently in the expressive style of the prompt.[8] Going one step further, [331] used a speech emotion conversion framework to modify the perceived emotion of a speech utterance while preserving its lexical content and speaker identity. Other studies have extended the idea of textless language processing or audio discrete representation to applications such as spoken question answering [333], speech separation [334], TTS [335], and speech-to-speech translation [336].

VII. Discussion and Conclusion

In this overview, we have presented the historical context of self-supervised learning and provided a thorough methodological review of important self-supervised speech representation models. Specifically, we have categorized the approaches into three categories, generative, contrastive and predictive, differing in terms of how the pretext task is defined. We have presented an overview of existing benchmarks and reviewed the efforts towards efficient zero-resource learning. Although the field is progressing rapidly, with new approaches reaching higher levels of performance, a couple of patterns have emerged: (1) The solid performance of Wav2vec 2.0 for speech recognition and many downstream tasks, as well as the public availability of its pre-train wide adoption in the community making it a "standard" go-to model. (2) The simplicity and stability of the HuBERT approach, as well as the resemblance of its training procedure to classic frame-level ASR systems, made it an easy choice for research extensions on improving representation quality, speech translation, and textless NLP.

Below we highlight various shortcomings of existing work and future research directions:

- **Using the representation model.** So far, there are two main ways to use representation models: Freeze the representation models and use them as feature extractors, or fine-tune the representation models on downstream tasks. Some efficient methods for leveraging SSL models exist in the NLP community. Adapters [337], [338], [339] are lightweight modules inserted into SSL models, and in downstream tasks, the parameters of SSL models are frozen, and only the adapters are trained. The prompt/instruction learning methods [20] also freeze the SSL parameters and control the output of SSL by adding additional information, which is called *prompt*, in the input. Both adapter-based methods and prompt/instruction learning yield competitive performance compared with fine-tuning in NLP applications, but there is only little re lated work for speech [340], [341]. In addition, prompt for speech SSL does not achieve comparable performance on sequence generation tasks like phoneme recognition and slot filling, so how to use prompt is still an open question.
- **Increasing the efficiency of the representation model.** As discussed in Section V-C, larger representation models lead to better downstream performance. Despite the success of these large models, they incur high costs in terms of memory and time for pre-training, fine-tuning, and even when used only to extract representations without gradient calculation. This makes them unsuitable for edge devices but also limits the ability to scale these models to very large datasets – and leads to a large energy consumption. Preliminary studies have been conducted on compressing speech representation models through network pruning [342] or knowledge distillation [343]. There has been quite some effort towards more efficient general neural network models via conditional computing [344] and neural network quantization [345] as well as extensive work on improving the specific efficiency of Transformer models, especially with the focus on self-attention [346], but these technology has not been widely used in speech SSL. Because speech is intrinsically represented as sequence, one way to reduce computation is to reduce the length of speech representation sequence but still keep the vital information in speech. But we have not been aware of any publication in this direction when writing this paper. On the other hand, non-streaming architectures in models such as the bidirectional Transformer have hindered the representation model used in streaming scenarios, leading to studies that address these problems [347]. We anticipate research in these directions to continue in the future.
- **Data-efficient approaches.** SOTA representation learning methods require large volumes of unlabeled speech during pre-training, going way beyond what babies need to understand language. Different learning approaches have different data needs, e.g., generative approaches could be more data efficient than contrastive or predictive approaches

[6][Online]. Available: https://speechbot.github.io/gslm/
[7][Online]. Available: https://speechbot.github.io/dgslm/
[8][Online]. Available: https://speechbot.github.io/pgslm/

since they are constrained by more bits of information to reconstruct their inputs. Comprehensive research is needed to study the data efficiency of different approaches.

- **Feature Disentanglement.** Speech SSL models show strengths on a surprisingly wide range of tasks [44], suggesting that representations contain different information. One way to further improve downstream tasks is to disentangle different information from the representation. For example, we can decompose the representation into content embedding and speaker embedding and use content embedding for ASR and speaker embedding for SID. Some work has been in this direction [348], [349], [350].
- **Creating robust models.** As discussed in Section V-D, studies have been conducted on the robustness of representation models [351]. However, the failure modes of SSL models are still poorly understood, and it remains unclear whether they provide more or less robustness to adversarial attacks than fully supervised models. Due to the importance of this research direction, while writing this paper, there is already some related research about enhancing the robustness of SSL models [281], [352], [353], [354] and identifying their vulnerability to adversarial attack [351].
- **Capturing higher-level semantic information.** Although many representation learning approaches can go beyond low-level phonetic modeling to capture some lexical information [355], they still struggle in higher-level semantic tasks easily captured by word-level counterparts like BERT. One workaround is two-stage training [77], [332]; however, this prevents propagating rich lexical and semantic knowledge modeled in the second stage to benefit the phonetically focused first stage.
- **Using text representation models to improve speech representation.** The amount of content information in speech corpora used to train speech representation models is far less than that of text representation models. Noting that the BERT training corpus exceeds 3 billion words [54], and assuming a typical speaking rate of 120 words per minute, a speech corpus containing the same content as the BERT training data would include 400,000 hours of audio, which exceeds the accumulated training data of all current speech representation models. Therefore, to enable speech representation models to better learn human language, for instance by extracting semantic information from acoustic signals, the use of text models such as BERT and GPT seems key: nevertheless, how to use these to improve speech representation model pre-training remains an open question.

We believe SSL representation models have considerable room to grow. The relationship between representation models and downstream tasks can be compared to the relationship between operating systems and applications. Today, even individuals can build applications with desired functions on a smartphone because the smartphone's operating system handles the complex communication with the hardware and provides a convenient developer interface. Likewise, as SSL representation models learn general knowledge from human speech, it is easy to develop new speech processing applications on this basis. From this viewpoint, speech representation models will play the role of operating systems in speech processing and further facilitate the continued development of speech technology.

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] G. Hinton et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.

[3] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, vol. 247. Berlin, Germany: Springer, 2012.

[4] T. Kemp and A. Waibel, "Unsupervised training of a speech recognizer: Recent experiments," in *Proc. Eur. Conf. Speech Commun. Technol.*, 1999, pp. 2725–2728.

[5] L. Lamel, J.-L. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Comput. Speech Lang.*, vol. 16, no. 1, pp. 115–129, 2002.

[6] J. Ma, S. Matsoukas, O. Kimball, and R. Schwartz, "Unsupervised training on large amounts of broadcast news data," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2006, pp. III–III.

[7] G. E. Hinton, "Learning multiple layers of representation," *Trends Cogn. Sci.*, vol. 11, pp. 428–434, 2007.

[8] Y. LeCun et al., "A tutorial on energy-based learning," in *Predicting Structured Data*. Cambridge, MA, USA: MIT Press, 2006.

[9] Y. Bengio, A. C. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[10] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[11] R. Gray, "Vector quantization," *IEEE ASSP Mag.*, vol. 1, no. 2, pp. 4–29, Apr. 1984.

[12] M. Jordan and R. Jacobs, "Hierarchical mixtures of experts and the EM algorithm," in *Proc. Int. Conf. Neural Netw.*, 1993,vol. 2, pp. 1339–1344.

[13] G. E. Hinton and R. Zemel, "Autoencoders, minimum description length and Helmholtz free energy," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 6, pp. 3–10, 1994.

[14] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[15] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[16] R. Bommasani et al., "On the opportunities and risks of foundation models," 2021, *arXiv:2108.07258*.

[17] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales, "Self-supervised representation learning: Introduction, advances and challenges," *IEEE Signal Process. Mag.*, vol. 39, no. 3, pp. 42–62, May 2022.

[18] X. Liu et al., "Self-supervised learning: Generative or contrastive," *IEEE Trans. Knowl. Data Eng.*, early access, doi: 10.1109/TKDE.2021.3090866.

[19] A. Rogers, O. Kovaleva, and A. Rumshisky, "A primer in BERTology: What we know about how BERT works," *Trans. Assoc. Comput. Linguistics*, vol. 8, pp. 842–866, 2020. [Online]. Available: https://aclanthology.org/2020.tacl-1.54

[20] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Comput. Surv.*, Aug., 2022.

[21] P. Xia, S. Wu, and B. Van Durme, "Which *BERT? A survey organizing contextualized encoders," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 7516–7533. [Online]. Available: https://aclanthology.org/2020.emnlp-main.608

[22] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, and X. Huang, "Pre-trained models for natural language processing: A survey," *Sci. China Technol. Sci.*, vol. 63, no. 10, pp. 1872–1897, Sep. 2020, doi: 10.1007/s11431-020-1647-3.

[23] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4037–4058, Nov. 2021.

[24] L. Borgholt, J. D. Havtorn, J. Edin, L. Maaløe, and C. Igel, "A brief overview of unsupervised neural speech representation learning," in *Proc. 2nd Workshop Self-supervised Learn. Audio Speech Process.*, 2022.

[25] S. Latif, R. Rana, S. Khalifa, R. Jurdak, J. Qadir, and B. W. Schuller, "Deep representation learning in speech processing: Challenges, recent advances, and future trends," 2021, *arXiv:2001.00378*.

[26] L. Rabiner and J. Wilpon, "Considerations in applying clustering techniques to speaker independent word recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1979, vol. 4, pp. 578–581.

[27] J. Wilpon and L. Rabiner, "A modified k-means clustering algorithm for use in isolated work recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 3, pp. 587–594, Jun. 1985.

[28] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains," *IEEE Speech Audio Process.*, vol. 2, no. 2, pp. 291–298, Apr. 1994.

[29] S. Young and P. Woodland, "State clustering in hidden Markov model-based continuous speech recognition," *Comput. Speech Lang.*, vol. 8, no. 4, pp. 369–383, 1994.

[30] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum mutual information estimation of hidden Markov model parameters for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1986, vol. 11, pp. 49–52.

[31] N. Smith and M. Gales, "Speech recognition using SVMs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 1197–1204.

[32] V. Venkataramani, S. Chakrabartty, and W. Byrne, "Support vector machines for segmental minimum Bayes risk decoding of continuous speech," in in *IEEE Workshop Autom. Speech Recognit. Understanding*, 2003, pp. 13–18.

[33] V. Wan and S. Renals, "SVMSVM: Support vector machine speaker verification methodology," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2003, pp. 221–224.

[34] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 788–798, May 2011.

[35] N. Dehak, P. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2011, pp. 857–860.

[36] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, 2010.

[37] M. U. Gutmann and A. Hyvärinen, "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics," *J. Mach. Learn. Res.*, vol. 13, no. 2, pp. 307–361, 2012.

[38] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, Jun. 1996.

[39] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. 19th Int. Conf. Neural Inf. Process. Syst.*, 2006, pp. 801–808.

[40] G. S. Sivaram, S. K. Nemala, M. Elhilali, T. D. Tran, and H. Hermansky, "Sparse coding for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2010, pp. 4346–4349.

[41] M. Ranzato, Y.-L. Boureau, S. Chopra, and Y. LeCun, "A unified energy-based framework for unsupervised learning," in *Proc. 11th Int. Conf. Artif. Intell. Statist.*, 2007, pp. 371–379.

[42] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Comput.*, vol. 14, no. 8, pp. 1771–1800, Aug. 2002.

[43] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "GLUE: A multi-task benchmark and analysis platform for natural language understanding," in *Proc. Int. Conf. Learn. Representations*, 2019.

[44] S. wen Yang et al., "SUPERB: Speech processing universal PERformance benchmark," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 1194–1198.

[45] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 649–666.

[46] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 132–149.

[47] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," 2015, *arXiv:1505.05192*.

[48] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Representations*, 2014.

[49] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1278–1286.

[50] L. Girin, S. Leglaive, X. Bie, J. Diard, T. Hueber, and X. Alameda-Pineda, "Dynamical variational autoencoders: A comprehensive review," *Found. Trends Mach. Learn.*, vol. 15, pp. 1–175, Dec. 2021.

[51] M. E. Peters et al., "Deep contextualized word representations," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2018, pp. 2227–2237.

[52] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[53] M. Shoeybi, M. Patwary, R. Puri, P. LeGresley, J. Casper, and B. Catanzaro, "Megatron-LM: Training multi-billion parameter language models using model parallelism," 2020, *arXiv:1909.08053*.

[54] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Hum. Lang. Technol.*, 2019, pp. 4171–4186.

[55] Y. Liu et al., "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.

[56] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2018, *arXiv:1807.03748*.

[57] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[58] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9729–9738.

[59] X. Chen, H. Fan, R. Girshick, and K. He, "Improved baselines with momentum contrastive learning," 2020, *arXiv:2003.04297*.

[60] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 9912–9924.

[61] S. H. K. Parthasarathi and N. Strom, "Lessons from building acoustic models with a million hours of speech," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 6670–6674.

[62] D. S. Park et al., "Improved noisy student training for automatic speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 2817–2821.

[63] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert, "Iterative pseudo-labeling for speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 741–745.

[64] A. Xiao et al., "Scaling ASR improves zero and few shot learning," in *Proc. Interspeech*, 2022, pp. 5135–5139.

[65] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[66] J. Cui et al., "Multilingual representations for low resource speech recognition and keyword search," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2015, pp. 259–266.

[67] P. Bell, J. Fainberg, O. Klejch, J. Li, S. Renals, and P. Swietojanski, "Adaptation algorithms for neural network-based speech recognition: An overview," *IEEE Open J. Signal Process.*, vol. 2, pp. 33–66, 2021.

[68] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2021, pp. 914–921.

[69] S. Pascual, M. Ravanelli, J. Serrà, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 161–165.

[70] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Proc. 29th Conf. Neural Inf. Process. Syst.*, Montréal, Quebec, Canada, 2015, pp. 2980–2988.

[71] M. Fraccaro, S. K. Sønderby, U. Paquet, and O. Winther, "Sequential neural models with stochastic layers," in *Proc. 30th Conf. Neural Inf. Process. Syst.*, Barcelona, Spain, 2016, pp. 2207–2215.

[72] W.-N. Hsu, Y. Zhang, and J. Glass, "Learning latent representations for speech generation and transformation," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 1273–1277.

[73] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1876–1887.

[74] E. Aksan and O. Hilliges, "STCN: Stochastic temporal convolutional networks," in *Proc. 7th Int. Conf. Learn. Representations*, New Orleans, LA, USA, 2019.

[75] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6309–6318.

[76] A. Polyak et al., "Speech resynthesis from discrete disentangled self-supervised representations," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 3615–3619.

[77] E. Kharitonov et al., "Text-free prosody-aware generative spoken language modeling," in *Proc. Assoc. Comput. Linguistics*, May 2022, pp. 8666–8681.

[78] K. Lakhotia et al., "On generative spoken language modeling from raw audio," *Trans. Assoc. Comput. Linguistics*, vol. 9, pp. 1336–1354, 2021.

[79] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," 2013, *arXiv:1308.3432*.

[80] A. v. d. Oord et al., "WaveNet: A generative model for raw audio," in *Proc. 9th ISCA Speech Synth. Workshop*, 2016, p. 125.

[81] D. K. Burton, J. E. Shore, and J. T. Buck, "A generalization of isolated word recognition using vector quantization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1983, pp. 1021–1024.

[82] F. Soong, A. Rosenberg, and L. R. B. Juang, "A vector quantization approach to speaker recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1985, pp. 387–390.

[83] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using WaveNet autoencoders," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 12, pp. 2041–2053, Dec. 2019.

[84] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel-softmax," in *Proc. Int. Conf. Learn. Representations*, 2017.

[85] R. Eloff et al., "Unsupervised acoustic unit discovery for speech synthesis using discrete latent-variable neural networks," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 1103–1107.

[86] M. D. Zeiler et al., "On rectified linear units for speech processing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2013, pp. 3517–3521.

[87] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "An auto-encoder based approach to unsupervised learning of subword units," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 7634–7638.

[88] L. Badino, A. Mereta, and L. Rosasco, "Discovering discrete subword units with binarized autoencoders and hidden-Markov-model encoders," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 3174–3178.

[89] H. Kamper, M. Elsner, A. Jansen, and S. Goldwater, "Unsupervised neural network based feature extraction using weak top-down constraints," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 5818–5822.

[90] D. Renshaw, H. Kamper, A. Jansen, and S. Goldwater, "A comparison of neural network methods for unsupervised representation learning on the zero resource speech challenge," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 3199–3203.

[91] S. Settle, K. Audhkhasi, K. Livescu, and M. Picheny, "Acoustically grounded word embeddings for improved acoustics-to-word speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 5641–5645.

[92] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An unsupervised autoregressive model for speech representation learning," in *Proc. Interspeech*, 2019, pp. 146–150.

[93] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 3497–3501.

[94] D. O'Shaughnessy, "Linear predictive coding," *IEEE Potentials*, vol. 7, no. 1, pp. 29–32, Feb. 1988.

[95] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural Comput.*, vol. 14, no. 4, pp. 715–770, 2002.

[96] J. S. Garofolo, "TIMIT acoustic-phonetic continuous speech corpus LDC93S1," *Linguistic Data Consortium*, 1993. [Online]. Available: https://catalog.ldc.upenn.edu/LDC93S1

[97] Y.-A. Chung and J. Glass, "Improved speech representations with multi-target autoregressive predictive coding," in *Proc. Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 2353–2358.

[98] Y.-A. Chung, H. Tang, and J. Glass, "Vector-quantized autoregressive predictive coding," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 3760–3764.

[99] S. Ling, Y. Liu, J. Salazar, and K. Kirchhoff, "Deep contextualized acoustic representations for semi-supervised speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 6429–6433.

[100] A. T. Liu, S.-w. Yang, P.-H. Chi, P.-c. Hsu, and H.-y. Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 6419–6423.

[101] D. Jiang et al., "Improving transformer-based speech recognition using unsupervised pre-training," 2019, *arXiv:1910.09932*.

[102] L. Liu and Y. Huang, "Masked pre-trained encoder base on joint CTC-Transformer," 2020, *arXiv:2005.11978*.

[103] W. Wang, Q. Tang, and K. Livescu, "Unsupervised pre-training of bidirectional speech encoders via masked reconstruction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 6889–6893.

[104] D. Jiang et al., "A further study of unsupervised pretraining for transformer based speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 6538–6542.

[105] X. Yue and H. Li, "Phonetically motivated self-supervised speech representation learning," *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 746–750.

[106] A. T. Liu, S.-W. Li, and H.-y. Lee, "TERA: Self-supervised learning of transformer encoder representation for speech," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 2351–2366, 2021.

[107] A. H. Liu, Y.-A. Chung, and J. Glass, "Non-autoregressive predictive coding for learning speech representations from local dependencies," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 3730–3734.

[108] Z. Yang et al., "Generalized Autoregressive Pretraining for Language Understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, 2019.

[109] X. Song, G. Wang, Y. Huang, Z. Wu, D. Su, and H. Meng, "Speech-XLNet: Unsupervised acoustic model pretraining for self-attention networks," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 3765–3769.

[110] S. Ling and Y. Liu, "DeCoAR 2.0: Deep contextualized acoustic representations with vector quantization," 2020, *arXiv:2012.06659*.

[111] J. Luo, J. Wang, N. Cheng, and J. Xiao, "Dropout regularization for self-supervised learning of transformer encoder speech representation," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 1169–1173.

[112] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.

[113] M. Ravanelli et al., "Multi-task self-supervised learning for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 6989–6993.

[114] Y.-A. Chung and J. Glass, "Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 811–815.

[115] M. Tagliasacchi, B. Gfeller, F. d. C. Quitry, and D. Roblek, "Self-supervised audio representation learning for mobile devices," 2019, *arXiv:1905.11796*.

[116] M. Tagliasacchi, B. Gfeller, F. de Chaumont Quitry, and D. Roblek, "Pre-training audio representations with self-supervision," *IEEE Signal Process. Lett.*, vol. 27, pp. 600–604, 2020.

[117] F. d. C. Quitry, M. Tagliasacchi, and D. Roblek, "Learning audio representations via phase prediction," 2019, *arXiv:1910.11910*.

[118] Y.-A. Chung, C.-C. Wu, C.-H. Shen, H.-Y. Lee, and L.-S. Lee, "Audio word2Vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2016, pp. 765–769.

[119] P.-H. Chi, P.-H. Chung, T.-H. Wu, C.-C. Hsieh, S.-W. Li, and H.-y. Lee, "Audio ALBERT: A lite BERT for self-supervised learning of audio representation," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2021, pp. 344–350.

[120] B. Milde and C. Biemann, "Unspeech: Unsupervised speech context embeddings," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 2693–2697.

[121] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Proc. Interspeech*, 2019, pp. 3465–3469.

[122] M. Riviere, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 7414–7418.

[123] K. Kawakami, L. Wang, C. Dyer, P. Blunsom, and A. van den Oord, "Learning robust and multilingual speech representations," in *Proc. Empirical Methods Natural Lang. Process.*, 2020, pp. 1182–1192.

[124] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *Proc. Int. Conf. Learn. Representations*, 2020.

[125] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 12449–12460.

[126] S. Sadhu et al., "wav2vec-C: A. self-supervised model for speech representation learning," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 711–715.

[127] Y. Chung et al., "w2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 244–250.

[128] D. Jiang, W. Li, M. Cao, W. Zou, and X. Li, "Speech SIMCLR: Combining contrastive and reconstruction objective for self-supervised speech representation learning," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 1544–1548.

[129] A. Baevski and A. Mohamed, "Effectiveness of Self-Supervised Pre-Training for ASRE," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 7694–7698.

[130] W.-N. Hsu, B. Bolte, Y. H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and M. Abdelrahman, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," in *Proc. Trans. Audio, Speech, Lang. Process.*, vol. 29, 2021, pp. 3451–3460.

[131] S. Chen et al., "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE J. Sel. Topics Signal Process.*, early access, Jul. 4, 2022, doi: 10.1109/JSTSP.2022.3188113.

[132] A. Baevski, W. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "data2vec: A general framework for self-supervised learning in speech, vision and language," 2022, *arXiv:2202.03555*.

[133] C.-C. Chiu, J. Qin, Y. Zhang, J. Yu, and Y. Wu, "Self-supervised learning with random-projection quantizer for speech recognition," in *Proc. Int. Conf. Mach. Learn.*, vol. 162, Jul. 17–23, 2022, pp. 3915–3924.

[134] M. Schultz and T. Joachims, "Learning a distance metric from relative comparisons," in *Proc. Neural Inf. Process. Syst.*, 2003, pp. 41–48.

[135] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," *Proc. Int. Conf. Artif. Intell. Statist.*, 2010, pp. 297–304.

[136] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.

[137] A. A. Razavi van den Oord and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," in *Proc, Adv. Neural Inf. Process. Syst.*, H. H. Wallach, A. Larochelle Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Red Hook, NY, USA: Curran Assoc., Inc., 2019.

[138] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, "Deep clustering for unsupervised learning of visual features," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 132–149.

[139] J. Grill et al., "Bootstrap Your Own Latent a New Approach to Self-Supervised Learning," in *Proc. Neural Inf. Process. Syst.*, 2020, Art. no. 1786.

[140] M. Caron et al., "Emerging properties in self-supervised vision transformers," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 9650–9660.

[141] R. B. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[142] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[143] E. D. Petajan, "Automatic lipreading to enhance speech recognition (speech reading)," Ph.D. dissertation, University of Illinois at Urbana-Champaign, Champaign, IL, USA, 1984.

[144] G. Potamianos, C. Neti, G. Gravier, A. Garg, and A. W. Senior, "Recent advances in the automatic recognition of audiovisual speech," *Proc. IEEE*, vol. 91, no. 9, pp. 1306–1326, Sep. 2003.

[145] P. S. Aleksic and A. K. Katsaggelos, "Audio-visual biometrics," *Proc. IEEE*, vol. 94, no. 11, pp. 2025–2044, Nov. 2006.

[146] M. Legerstee, "Infants use multimodal information to imitate speech sounds," *Infant Behav. Develop.*, vol. 13, no. 3, pp. 343–354, 1990.

[147] D. Roy, "Learning from sights and sounds: A computational model," Ph.D. dissertation, MIT Media Lab., Cambridge, MA, USA, 1999.

[148] B. Lee et al., "AVICAR: Audio-visual speech corpus in a car environment," in *Proc. 8th Int. Conf. Spoken Lang. Process.*, 2004, pp. 2489–2492.

[149] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *Proc. Asian Conf. Comput. Vis.*, 2016, pp. 87–103.

[150] B. Shi, W.-N. Hsu, K. Lakhotia, and A. Mohamed, "Learning audio-visual speech representation by masked multimodal cluster prediction," in *Proc. Int. Conf. Learn. Representations*, 2022.

[151] J. Westbury, P. Milenkovic, G. Weismer, and R. Kent, "X-ray microbeam speech production database," *J. Acoust. Soc. Amer.*, vol. 88, no. S1, pp. S56–S56, 1990.

[152] A. Wrench, "A new resource for production modelling in speech technology," *Inst. Acoust.*, vol. 23, no. 3, pp. 207–218, 2001.

[153] S. Narayanan et al., "A multimodal real-time MRI articulatory corpus for speech research," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2011, pp. 1556–1560.

[154] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 689–696.

[155] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "Integrating articulatory data in deep neural network-based acoustic modeling," *Comput. Speech Lang.*, vol. 36, pp. 173–195, 2016.

[156] N. Srivastava and R. R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 2222–2230.

[157] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[158] G. Andrew, R. Arora, J. A. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 1247–1255.

[159] W. Wang, R. Arora, K. Livescu, and J. A. Bilmes, "On deep multi-view representation learning," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1083–1092.

[160] W. Wang, X. Yan, H. Lee, and K. Livescu, "Deep variational canonical correlation analysis," 2016, *arXiv:1610.03454*.

[161] T. Michaeli, W. Wang, and K. Livescu, "Nonparametric canonical correlation analysis," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1967–1976.

[162] T. Melzer, M. Reiter, and H. Bischof, "Nonlinear feature extraction using generalized canonical correlation analysis," in *Proc. Int. Conf. Artif. Neural Netw.*, 2001, pp. 353–360.

[163] P. L. Lai and C. Fyfe, "Kernel and nonlinear canonical correlation analysis," *Int. J. Neural Syst.*, vol. 10, no. 5, pp. 365–377, 2000.

[164] P. L. Lai and C. Fyfe, "A neural implementation of canonical correlation analysis," *Neural Netw.*, vol. 12, no. 10, pp. 1391–1397, 1999.

[165] F. R. Bach and M. I. Jordan, "A probabilistic interpretation of canonical correlation analysis," Dept. Statist., Univ. California, Berkeley, Tech. Rep. 688, 2005.

[166] W. Wang, R. Arora, K. Livescu, and J. Bilmes, "Unsupervised learning of acoustic features via deep canonical correlation analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 4590–4594.

[167] K. M. Hermann and P. Blunsom, "Multilingual distributed representations without word alignment," in *Proc. Int. Conf. Learn. Representations*, 2013.

[168] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, and L. Heck, "Learning deep structured semantic models for web search using clickthrough data," in *Proc. Int. Conf. Inf. Knowl. Manage.*, 2013, pp. 2333–2338.

[169] B. Shi, W.-N. Hsu, and A. Mohamed, "Robust self-supervised audio-visual speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 2118–2122.

[170] G. Chrupała, "Visually grounded models of spoken language: A survey of datasets, architectures and evaluation techniques," *J. Artif. Intell. Res.*, vol. 73, 2022, pp. 673–707.

[171] G. Synnaeve, M. Versteegh, and E. Dupoux, "Learning words from images and speech," in *Proc. Proc. Adv. Neural Inf. Process. Syst. Workshop Learn. Semantics*, 2014.

[172] D. Harwath, A. Torralba, and J. R. Glass, "Unsupervised learning of spoken language with visual context," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1866–1874.

[173] D. Harwath and J. R. Glass, "Deep multimodal semantic embeddings for speech and images," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2015, pp. 237–244.

[174] D. Merkx, S. Frank, and M. Ernestus, "Language learning using speech to image retrieval," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 1841–1845.

[175] A. Rouditchenko et al., "AVLnet: Learning audio-visual language representations from instructional videos," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 1584–1588.

[176] P. Peng and D. Harwath, "Fast-slow transformer for visually grounding speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 7727–7731.

[177] G. Ilharco, Y. Zhang, and J. Baldridge, "Large-scale representation learning from visually grounded untranscribed speech," in *Proc. 23rd Conf. Comput. Natural Lang. Learn.*, 2019, pp. 55–65.

[178] R. Sanabria, A. Waters, and J. Baldridge, "Talk, don't write: A study of direct speech-based image retrieval," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 2976–2980.

[179] D. M. Chan, S. Ghosh, D. Chakrabarty, and B. Hoffmeister, "Multi-modal pre-training for automated speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 246–250.

[180] P. Peng and D. Harwath, "Self-supervised representation learning for speech using visual grounding and masked language modeling," in *Proc. AAAI Conf. Artif. Intell. SAS Workshop*, 2022.

[181] D. Harwath, W.-N. Hsu, and J. Glass, "Learning hierarchical discrete linguistic units from visually-grounded speech," in *Proc. Int. Conf. Learn. Representations*, 2019.

[182] G. Chrupała, L. Gelderloos, and A. Alishahi, "Representations of language in a model of visually grounded speech signal," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 613–622.

[183] O. Scharenborg et al., "Linguistic unit discovery from multi-modal inputs in unwritten languages: Summary of the "speaking rosetta" JSALT 2017 workshop," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 4979–4983.

[184] D. Harwath, A. Recasens, D. Surís, G. Chuang, A. Torralba, and J. Glass, "Jointly discovering visual objects and spoken words from raw sensory input," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 649–665.

[185] P. Peng and D. Harwath, "Word discovery in visually grounded, self-supervised speech models," in *Proc. Interspeech*, 2022, pp. 2823–2827.

[186] L. Wang and M. Hasegawa-Johnson, "A DNN-HMM-DNN hybrid model for discovering word-like units from spoken captions and image regions," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 1456–1460.

[187] D. Harwath, G. Chuang, and J. Glass, "Vision as an interlingua: Learning multilingual semantic embeddings of untranscribed speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 4969–4973.

[188] W. N. Havard, J.-P. Chevrot, and L. Besacier, "Models of visually grounded speech signal pay attention to nouns: A bilingual experiment on english and japanese," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 8618–8622.

[189] H. Kamper and M. Roth, "Visually grounded cross-lingual keyword spotting in speech," in *Proc. 6th Int. Workshop Spoken Lang. Technol. Under-Resourced Lang.*, 2018, pp. 248–252.

[190] A. Pasad, B. Shi, H. Kamper, and K. Livescu, "On the contributions of visual and textual supervision in low-resource semantic speech retrieval," in *Proc. Interspeech*, 2019, pp. 4195–4199.

[191] A. Bapna et al., "SLAM: A unified encoder for speech and language modeling via speech-text joint pre-training," 2021, *arXiv:2110.10329*.

[192] K. Levin, A. Jansen, and B. Van Durme, "Segmental acoustic indexing for zero resource keyword search," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 5828–5832.

[193] G. Chen, C. Parada, and T. N. Sainath, "Query-by-example keyword spotting using long short-term memory networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 5236–5240.

[194] S. Settle, K. Levin, H. Kamper, and K. Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 2874–2878.

[195] H. Kamper, K. Livescu, and S. Goldwater, "An embedded segmental k-means model for unsupervised segmentation and clustering of speech," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2017, pp. 719–726.

[196] H. Kamper, A. Jansen, and S. Goldwater, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *Comput. Speech Lang.*, vol. 46, pp. 154–174, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0885230816301905

[197] A. L. Maas, S. D. Miller, T. M. O'neil, A. Y. Ng, and P. Nguyen, "Word-level acoustic modeling with convolutional vector regression," in *Proc. Int. Conf. Mach. Learn. Workshop Representation Learn.*, 2012.

[198] S. Bengio and G. Heigold, "Word embeddings for speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2014, pp. 1053–1057.

[199] K. Levin, K. Henry, A. Jansen, and K. Livescu, "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2013, pp. 410–415.

[200] N. Holzenberger, M. Du, J. Karadayi, R. Riad, and E. Dupoux, "Learning word embeddings: Unsupervised methods for fixed-size representations of variable-length speech segments," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 2683–2687.

[201] H. Kamper, "Truly unsupervised acoustic word embeddings using weak top-down constraints in encoder-decoder models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 6535–3539.

[202] P. Peng, H. Kamper, and K. Livescu, "A correspondence variational autoencoder for unsupervised acoustic word embeddings," in *Proc. NeurIPS Workshop Self-Supervised Learn. Speech Audio Process.*, 2020.

[203] M. A. Carlin, S. Thomas, A. Jansen, and H. Hermansky, "Rapid evaluation of speech representations for spoken term discovery," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2011, pp. 821–824.

[204] H. Kamper, W. Wang, and K. Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2016, pp. 4950–4954.

[205] S. Toshniwal, H. Shi, B. Shi, L. Gao, K. Livescu, and K. Gimpel, "A cross-task analysis of text span representations," in *Proc. 5th Workshop Representation Learn.*, 2020, pp. 166–176.

[206] S. Wang, L. Thompson, and M. Iyyer, "Phrase-BERT: Improved phrase embeddings from BERT with an application to corpus exploration," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2021, pp. 10837–10851.

[207] L. van Staden and H. Kamper, "A comparison of self-supervised speech representations as input features for unsupervised acoustic word embeddings," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2021, pp. 927–934.

[208] J. Kahn et al., "Libri-light: A benchmark for ASR with limited or no supervision," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 7669–7673.

[209] J. F. Gemmeke et al., "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 776–780.

[210] A. Ephrat et al., "Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–11, 2018.

[211] C. Cieri, D. Miller, and K. Walker, "The fisher corpus: A resource for the next generations of speech-to-text," in *Proc. Int. Conf. Lang. Resour. Eval.*, 2004, vol. 4, pp. 69–71.

[212] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 5206-5210.

[213] D. B. Paul and J. Baker, "The design for the wall street journal-based CSR corpus," in *Proc. Speech Natural Lang.: Proc. Workshop*, Harriman, NY, USA, 1992, pp. 357–362.

[214] R. Ardila et al., "Common voice: A massively-multilingual speech corpus," in *Proc. Int. Conf. Lang. Resour. Eval.*, 2020, pp. 4218–4222.

[215] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "MLS: A large-scale multilingual dataset for speech research," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 2757–2761.

[216] C. Wang et al., "VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," in *Proc. 59th Ann. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 993–1003.

[217] M. J. Gales, K. M. Knill, A. Ragni, and S. P. Rath, "Speech recognition and keyword spotting for low-resource languages: BABEL project research at CUED," in *Proc. 4th Int. Workshop Spoken Lang. Technol. Under-Resourced Lang.*, 2014, pp. 16–23.

[218] A. Babu et al., "XLS-R: Self-supervised cross-lingual speech representation learning at scale," in *Proc. Interspeech*, 2022, pp. 2278–2282.

[219] S. Chen et al., "UniSpeech-SAT: Universal speech representation learning with speaker aware pre-training," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 6152–6156.

[220] G. Chen et al., "GigaSpeech: An evolving, multi-domain ASR corpus with 10,000 hours of transcribed audio," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 3670–3674.

[221] F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Esteve, "TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation," in *Proc. Int. Conf. Speech Comput.*, 2018, pp. 198–208.

[222] A. Rousseau, P. Deléglise, and Y. Esteve, "TED-LIUM: An automatic speech recognition dedicated corpus," in *Proc. Int. Conf. Lang. Resour. Eval.*, 2012, pp. 125–129.

[223] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1992, pp. 517–520.

[224] J. Valk and T. Alumäe, "VoxLingua107: A dataset for spoken language recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2021, pp. 652–658.

[225] Y. Liu, P. Fung, Y. Yang, C. Cieri, S. Huang, and D. Graff, "HKUST/MTS: A very large scale mandarin telephone speech corpus," in *Proc. Int. Symp. Chin. Spoken Lang. Process.*, 2006, pp. 724–735.

[226] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "AISHELL-1: An open-source mandarin speech corpus and a speech recognition baseline," in *Proc. 20th Conf. Oriental Chapter Int. Coordinating Committee Speech Databases Speech I/O Syst. Assessment*, 2017, pp. 1–5.

[227] Beijing DataTang Technology Co., Ltd., "Aidatatang_200zh, a free Chinese mandarin speech corpus." [Online]. Available: http://www.datatang.com

[228] Magic Data Technology Co., Ltd., "MAGICDATA mandarin chinese read speech corpus," 2019. [Online]. Available: http://www.imagicdatatech.com/index.php/home/dataopensource/data_info/id/101

[229] Surfingtech, "ST-CMDS-20170001_1 Free ST Chinese mandarin corpus."

[230] Primewords Information Technology Co., Ltd., "Primewords Chinese corpus set 1," 2018. [Online]. Available: https://www.primewords.cn

[231] M. Ravanelli, L. Cristoforetti, R. Gretter, M. Pellin, A. Sosi, and M. Omologo, "The DIRHA-English corpus and related tasks for distant-speech recognition in domestic environments," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2015, pp. 275–282.

[232] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 1561–1565.

[233] V. Hozjan, Z. Kacic, A. Moreno, A. Bonafonte, and A. Nogueiras, "Interface databases: Design and collection of a multilingual emotional speech database," in *Proc. Int. Conf. Lang. Resour. Eval.*, 2002, pp. 2024–2028.

[234] A. B. Zadeh, P. P. Liang, S. Poria, E. Cambria, and L.-P. Morency, "Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph," in *Proc. Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 2236–2246.

[235] C. Veaux, J. Yamagishi, and K. MacDonald, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit," 2017.

[236] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 2616–2620.

[237] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 814–818.

[238] X. Anguera, L.-J. Rodriguez-Fuentes, A. Buzo, F. Metze, I. Szöke, and M. Penagarikano, "QUESST2014: Evaluating query-by-example speech search in a zero-resource setting with real-life queries," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 5833–5837.

[239] A. C. Kocabiyikoglu, L. Besacier, and O. Kraif, "Augmenting LibriSpeech with french translations: A multimodal corpus for direct speech translation evaluation," in *Proc. Int. Conf. Lang. Resour. Eval.*, 2018, pp. 1–5.

[240] C. Wang, A. Wu, and J. Pino, "CoVoST 2 and massively multilingual speech-to-text translation," in *Proc. Interspeech*, 2021, pp. 2247–2251.

[241] M. Y. Tachbelie, S. T. Abate, and L. Besacier, "Using different acoustic, lexical and language modeling units for ASR of an under-resourced language–Amharic," *Speech Commun.*, vol. 56, pp. 181–194, 2014.

[242] F. A. A. Laleye, L. Besacier, E. C. Ezin, and C. Motamed, "First automatic fongbe continuous speech recognition system: Development of acoustic models and language models," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, 2016, pp. 477–482.

[243] H. Gelas, L. Besacier, and F. Pellegrino, "Developments of swahili resources for an automatic speech recognition system," in *Proc. Spoken Lang. Technol. Under-Resourced Lang.*, 2012, pp. 94–101.

[244] E. Gauthier, L. Besacier, S. Voisin, M. Melese, and U. P. Elingui, "Collecting resources in sub-Saharan African languages for automatic speech recognition: A case study of Wolof," in *Proc. 10th Lang. Resour. Eval. Conf.*, 2016, pp. 3863–3867.

[245] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," 2015, *arXiv:1510.08484*.

[246] D. Stowell, M. D. Wood, H. Pamuła, Y. Stylianou, and H. Glotin, "Automatic acoustic detection of birds through deep learning: The first bird audio detection challenge," *Methods Ecol. Evol.*, vol. 10, no. 3, pp. 368–380, 2019.

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

[248] T. Oponowicz, "Spoken language identification," 2018. [Online]. Available: https://www.kaggle.com/toponowicz/spoken-language-identification

[249] A. Mesaros, T. Heittola, and T. Virtanen, "Detection and classification of acoustic scenes and events," in *Proc. Workshop Detection and Classification Acoust, Scenes Events*, 2018, pp. 9–13.

[250] J. Engel et al., "Neural audio synthesis of musical notes with WaveNet autoencoders," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1068–1077.

[251] M. Ravanelli and Y. Bengio, "Learning speaker representations with mutual information," in *Proc. Interspeech*, 2019, pp. 1153–1157.

[252] S. Khurana et al., "A convolutional deep Markov model for unsupervised speech representation learning," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 3790–3794.

[253] S. Weinberger, "Speech Accent Archive," George Mason University, Manassas, Virginia, 2015.

[254] T. Likhomanenko, Q. Xu, J. Kahn, G. Synnaeve, and R. Collobert, "slimIPL: Language-model-Free iterative pseudo-labeling," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021.

[255] Y. Zhang et al., "Pushing the limits of semi-supervised learning for automatic speech recognition," in *Proc. Workshop Self-Supervised Learn. Speech Audio Process.*, 2020.

[256] M. Hajibabaei and D. Dai, "Unified hypersphere embedding for speaker recognition," 2018, *arXiv:1807.08312*.

[257] A. Hajavi and A. Etemad, "Siamese capsule network for end-to-end speaker recognition in the wild," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 7203–7207.

[258] Y. Wang, A. Boumadane, and A. Heba, "A fine-tuned wav2vec 2.0/HuBERT benchmark for speech emotion recognition, speaker verification and spoken language understanding," 2021, *arXiv:2111.02735*.

[259] L. J. Rodríguez-Fuentes, A. Varona, M. Penagarikano, G. Bordel, and M. Diez, "GTTS-EHU systems for QUESST at MediaEval 2014," in *Proc. MediaEval*, 2014.

[260] S. Evain et al., "LeBenchmark: A. reproducible framework for assessing self-supervised representation learning from speech," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 1439–1443.

[261] E. Dunbar et al., "The zero resource speech challenge 2020: Discovering discrete subword and word units," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 4832–4835.

[262] J. Turian et al., "Hear: Holistic evaluation of audio representations," in *Proc. NeurIPS Competitions Demonstrations Track*, 2022, vol. 176, pp. 125–145.

[263] J. Shor et al., "Towards learning a universal non-semantic representation of speech," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 140–144.

[264] L. Wang et al., "Towards learning universal audio representations," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 4593–4597.

[265] A. Tjandra, S. Sakti, and S. Nakamura, "Transformer VQ-VAE for unsupervised unit discovery and speech synthesis: ZeroSpeech 2020 challenge," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 4851–4855.

[266] B. van Niekerk, L. Nortje, and H. Kamper, "Vector-quantized neural networks for acoustic unit discovery in the ZeroSpeech 2020 challenge," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 4836–4840.

[267] M. Faruqi and C. Dyer, "Community evaluation and exchange of word vectors and wordvectors.org," in *Proc. 52nd Annu. Meeting Assoc. Comput. Linguistics: Syst. Demonstration*, 2014, pp. 19–24.

[268] A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, "Unsupervised speech recognition," in *Proc. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 27826–27839.

[269] E. Voita, R. Sennrich, and I. Titov, "The bottom-up evolution of representations in the Transformer: A. study with machine translation and language modeling objectives," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 4396–4406.

[270] T. G. Grigg, D. Busbridge, J. Ramapuram, and R. Webb, "Do self-supervised and supervised methods learn similar visual representations?," 2021. [Online]. Available: https://arxiv.org/abs/2110.00528

[271] B. van Niekerk, L. Nortje, M. Baas, and H. Kamper, "Analyzing speaker information in self-supervised models to improve zero-resource speech processing," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 1554–1558.

[272] S. Ling, J. Salazar, Y. Liu, and K. Kirchhoff, "BERTphone: Phonetically-aware encoder representations for utterance-level speaker and language recognition," in *Proc. Odyssey: Speaker Lang. Recognit. Workshop*, 2020, pp. 9–16.

[273] C. Wang et al., "Improving self-supervised learning for speech recognition with intermediate layer supervision," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 7092–7096.

[274] S. wen Yang, A. T. Liu, and H. yi Lee, "Understanding self-attention of self-supervised audio transformers," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 3785–3789.

[275] Y.-A. Chung, Y. Belinkov, and J. Glass, "Similarity analysis of self-supervised speech representations," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 3040–3044.

[276] H. Zhou, A. Baevski, and M. Auli, "A comparison of discrete latent variable models for speech representation learning," 2020. [Online]. Available: https://arxiv.org/abs/2010.14230

[277] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 7414–7418.

[278] J. Pu, Y. Yang, R. Li, O. Elibol, and J. Droppo, "Scaling effect of self-supervised models," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 1084–1088.

[279] M. Versteegh et al., "The zero resource speech challenge 2015," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 3169–3173.

[280] X. Chang et al., "An exploration of self-supervised pretrained representations for end-to-end speech recognition," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 228–235.

[281] W.-N. Hsu et al., "Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 721–725.

[282] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," in *Proc. Interspeech*, 2021, pp. 2426–2430.

[283] D.-R. Liu, K.-Y. Chen, H.-Y. Lee, and L.-s. Lee, "Completely unsupervised phoneme recognition by adversarially learning mapping relationships from audio embeddings," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 3748–3752.

[284] Y.-H. Wang, H. yi Lee, and L. shan Lee, "Segmental audio word2Vec: Representing utterances as sequences of vectors with applications in spoken term detection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 6269–6273.

[285] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein GANs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5769–5779.

[286] Y.-A. Chung, W.-H. Weng, S. Tong, and J. Glass, "Unsupervised cross-modal alignment of speech and text embedding spaces," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31, pp. 7354–7364.

[287] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou, "Word translation without parallel data," in *Proc. Int. Conf. Learn. Representations*, 2018.

[288] Y.-A. Chung, W.-H. Weng, S. Tong, and J. Glass, "Towards unsupervised speech-to-text translation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 7170–7174.

[289] M. Artetxe, G. Labaka, and E. Agirre, "A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, Jul. 2018, pp. 789–798. [Online]. Available: https://aclanthology.org/P18-1073

[290] C.-K. Yeh, J. Chen, C. Yu, and D. Yu, "Unsupervised speech recognition via segmental empirical output distribution matching," in *Proc. Int. Conf. Learn. Representations*, 2018.

[291] Y.-H. Wang, C.-T. Chung, and H.-y. Lee, "Gate activation signal analysis for gated recurrent neural networks and its correlation with phoneme boundaries," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2017, pp. 3822–3826.

[292] Y. Liu, J. Chen, and L. Deng, "Unsupervised sequence classification using sequential output statistics," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3553–3562.

[293] K.-Y. Chen, C.-P. Tsai, D.-R. Liu, H.-Y. Lee, and L.-s. Lee, "Completely unsupervised speech recognition by a generative adversarial network harmonized with iteratively refined hidden Markov models," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 1856–1860.

[294] O. Klejch, E. Wallington, and P. Bell, "Deciphering speech: A zero-resource approach to cross-lingual transfer in ASR," in *Proc. Interspeech*, 2022, pp. 2288–2292.

[295] S. Ravi and K. Knight, "Deciphering foreign language," in *Proc. 49th Annu. Meeting Assoc. Comput. Linguistics: Hum. Lang. Technol.*, Portland, OR, USA, 2011, pp. 12–21. [Online]. Available: https://aclanthology.org/P11-1002

[296] A. H. Liu, W.-N. Hsu, M. Auli, and A. Baevski, "Towards end-to-end unsupervised speech recognition," 2022, *arXiv:2204.02492*.

[297] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[298] M. Arjovsky, S. Chintala, and L. Bottou, "GAN wasserstein," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[299] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, "Unsupervised neural machine translation," in *Proc. Int. Conf. Learn. Representations*, 2018.

[300] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," *Proc. Int. Conf. Learn. Representations*, 2018.

[301] G.-T. Lin, C.-J. Hsu, D.-R. Liu, H.-Y. Lee, and Y. Tsao, "Analyzing the robustness of unsupervised speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 8202–8206.

[302] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2017, pp. 301–308.

[303] T. Hori, R. Astudillo, T. Hayashi, Y. Zhang, S. Watanabe, and J. Le Roux, "Cycle-consistency training for end-to-end speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 6271–6275.

[304] G. Wang et al., "Improving speech recognition using consistent predictions on synthesized speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 7029–7033.

[305] A. Tjandra, S. Sakti, and S. Nakamura, "Machine speech chain with one-shot speaker adaptation," in *Proc. Interspeech*, 2018, pp. 887–891.

[306] A. Tjandra, S. Sakti, and S. Nakamura, "End-to-end feedback loss in speech chain framework via straight-through estimator," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 6281–6285.

[307] M. K. Baskar, S. Watanabe, R. Astudillo, T. Hori, L. Burget, and J. Černockỳ, "Semi-supervised sequence-to-sequence ASR using unpaired speech and text," in *Proc. Interspeech*, 2019, pp. 3790–3794.

[308] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Mach. Learn.*, vol. 8, no. 3, pp. 229–256, 1992.

[309] F. Jelinek, *Statistical Methods for Speech Recognition*. Cambridge, MA, USA: MIT Press, 1997.

[310] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Comput. Speech Lang.*, vol. 16, no. 1, pp. 69–88, 2002.

[311] C. Gulcehre et al., "On using monolingual corpora in neural machine translation," 2015, *arXiv:1503.03535*.

[312] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," 2016, *arXiv:1612.02695*.

[313] J. Shen et al., "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2018, pp. 4779–4783.

[314] J. Li, R. Gadde, B. Ginsburg, and V. Lavrukhin, "Training neural speech recognition systems with synthetic speech augmentation," 2018, *arXiv:1811.00707*.

[315] S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, "Multi-speaker sequence-to-sequence speech synthesis for data augmentation in acoustic-to-word speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 6161–6165.

[316] A. Rosenberg et al., "Speech recognition with augmented synthesized speech," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2019, pp. 996–1002.

[317] A. Laptev, R. Korostik, A. Svischev, A. Andrusenko, I. Medennikov, and S. Rybin, "You do not need more data: Improving end-to-end speech recognition by text-to-speech data augmentation," in *Proc. 13th Int. Congr. Image, Signal Process., Biomed. Eng. Inform.*, 2020, pp. 439–444.

[318] Y. Huang, L. He, W. Wei, W. Gale, J. Li, and Y. Gong, "Using personalized speech synthesis and neural language generator for rapid speaker adaptation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 7399–7403.
[319] T. Hayashi et al., "Back-translation-style data augmentation for end-to-end ASR," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2018, pp. 426–433.
[320] A. Renduchintala, S. Ding, M. Wiesner, and S. Watanabe, "Multi-modal data augmentation for end-to-end ASR," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2018, pp. 2394–2398.
[321] R. Masumura, N. Makishima, M. Ihori, A. Takashima, T. Tanaka, and S. Orihashi, "Phoneme-to-grapheme conversion based large-scale pre-training for end-to-end automatic speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2020, pp. 2822–2826.
[322] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, 2016, pp. 86–96.
[323] A. Jansen et al., "A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 8111–8115.
[324] E. Dunbar et al., "The zero resource speech challenge 2017," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2017, pp. 323–330.
[325] E. Dunbar et al., "The zero resource speech challenge 2019: TTS without T," in *Proc. Interspeech*, 2019, pp. 1088–1092.
[326] T. A. Nguyen et al., "The zero resource speech benchmark 2021: Metrics and baselines for unsupervised spoken language modeling," 2020, *arXiv:2011.11588*.
[327] L. Ondel, L. Burget, and J. Černockỳ, "Variational inference for acoustic unit discovery," *Procedia Comput. Sci.*, vol. 81, pp. 80–86, 2016.
[328] M. Heck, S. Sakti, and S. Nakamura, "Feature optimized DPGMM clustering for unsupervised subword modeling: A contribution to ZeroSpeech 2017," in *Proc. IEEE Workshop Autom. Speech Recognit. Understanding*, 2017, pp. 740–746.
[329] A. Tjandra, B. Sisman, M. Zhang, S. Sakti, H. Li, and S. Nakamura, "VQ-VAE unsupervised unit discovery and multi-scale Code2Spec inverter for ZeroSpeech challenge 2019," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 1118–1122.
[330] T. Maekaku, X. Chang, Y. Fujita, L.-W. Chen, S. Watanabe, and A. Rudnicky, "Speech representation learning combining conformer CPC with deep cluster for the ZeroSpeech challenge 2021," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 1564–1568.
[331] F. Kreuk et al., "Textless speech emotion conversion using decomposed and discrete representations," 2021, *arXiv:2111.07402*.
[332] T. A. Nguyen et al., "Generative spoken dialogue language modeling," 2022, *arXiv:2203.16502*.
[333] G.-T. Lin et al., "DUAL: Discrete spoken unit adaptive learning for textless spoken question answering," in *Proc. Interspeech*, 2022, pp. 5165–5169.
[334] J. Shi, X. Chang, T. Hayashi, Y.-J. Lu, S. Watanabe, and B. Xu, "Discretization and re-synthesis: An alternative method to solve the cocktail party problem," 2021, *arXiv:2112.09382*.
[335] T. Hayashi and S. Watanabe, "Discretalk: Text-to-speech as a machine translation problem," 2020, *arXiv:2005.05525*.
[336] A. Lee et al., "Textless speech-to-speech translation on real data," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics*, 2022, pp. 860–872.
[337] N. Houlsby et al., "Parameter-efficient transfer learning for NLP," *Proc. Int. Conf. Mach. Learn. Res.*, vol. 97, pp. 2790–2799, Jun. 9–15, 2019.
[338] E. B. Zaken, S. Ravfogel, and Y. Goldberg, "BitFit: Simple parameter-efficient fine-tuning for Transformer-based masked language-models," in *Proc. Assoc. Comput. Linguistics*, 2021, pp. 1–9.
[339] D. Guo, A. Rush, and Y. Kim, "Parameter-efficient transfer learning with diff pruning," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguistics, 11th Int. Joint Conf. Natural Lang. Process.*, 2021, pp. 4884–4896. [Online]. Available: https://aclanthology.org/2021.acl-long.378
[340] B. Thomas, S. Kessler, and S. Karout, "Efficient adapter transfer of self-supervised speech models for automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 7102–7106.
[341] K.-W. Chang, W.-C. Tseng, S.-W. Li, and H. yi Lee, "SpeechPrompt: An exploration of prompt tuning on generative spoken language model for speech processing tasks," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 5005–5009.
[342] C.-I. J. Lai et al., "PARP: Prune, adjust and re-prune for self-supervised speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 21256–21272.
[343] H.-J. Chang, S. wen Yang, and H. yi Lee, "DistilHuBERT: Speech representation learning by layer-wise distillation of hidden-unit BERT," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 7087–7091.
[344] E. Bengio, P.-L. Bacon, J. Pineau, and D. Precup, "Conditional computation in neural networks for faster models," Jan. 2016, *arXiv:1511.06297*.
[345] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," *Low-Power Comput. Vis.: Improving Efficiency Artif. Intell.*, Jun. 2021.
[346] Y. Tay, M. Dehghani, D. Bahri, and D. Metzler, "Efficient transformers: A survey," *ACM Comput. Surv.*, Apr. 2022.
[347] S. Cao et al., "Improving streaming transformer based ASR under a framework of self-supervised learning," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2021, pp. 706–710.
[348] K. Qian et al., "ContentVec: An improved self-supervised speech representation by disentangling speakers," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 18003–18017.
[349] H.-S. Choi, J. Lee, W. Kim, J. H. Lee, H. Heo, and K. Lee, "Neural analysis and synthesis: Reconstructing speech from self-supervised representations," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2021, pp. 16251–16265.
[350] D. M. Chan and S. Ghosh, "Content-context factorized representations for automated speech recognition," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 61–65.
[351] H. Wu, B. Zheng, X. Li, X. Wu, H. yi Lee, and H. Meng, "Characterizing the adversarial vulnerability of speech self-supervised learning," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 3164–3168.
[352] K. P. Huang, Y.-K. Fu, Y. Zhang, and H.-y. Lee, "Improving distortion robustness of self-supervised speech processing tasks with domain adaptation," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, 2022, pp. 2193–2197.
[353] H. Wang et al., "Improving noise robustness of contrastive speech representation learning with speech reconstruction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 6062–6066.
[354] Q.-S. Zhu, J. Zhang, Z.-Q. Zhang, M.-H. Wu, X. Fang, and L.-R. Dai, "A noise-robust self-supervised pre-training model based speech representation learning for automatic speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 3174–3178.
[355] T. A. Nguyen, B. Sagot, and E. Dupoux, "Are discrete units necessary for spoken language modeling?," 2022, *arXiv:2203.05936*.

Abdelrahman Mohamed (Member, IEEE) is a Research Scientist in the Fundamental AI Research group (FAIR), Meta. He was a Principal Scientist/Manager with Amazon Alexa and a Researcher at Microsoft Research. He was in the team that started the Deep Learning revolution in spoken language processing in 2009. He received the IEEE Signal Processing Society Best Paper Award in 2016. He focuses on improving and benchmarking speech representation learning, where he co-organized multiple workshops and tutorials.

Hung-yi Lee is currently an Associate Professor with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, with a joint appointment with the Department of Computer Science and Information Engineering. He is also the Co-organizer of the special session on *New Trends in Self-Supervised Speech Processing* at Interspeech (2020) and the workshop on *Self-Supervised Learning for Speech and Audio Processing* at NeurIPS (2020) and AAAI (2022).

Lasse Borgholt received the Ph.D. degree from the University of Copenhagen, Copenhagen, Denmark, in 2022. He was supervised by Professors Christian Igel and Anders Søgaard, and an Adjunct Associate Professor Lars Maaløe. He is currently an Industrial Machine Learning Researcher with the Danish Company Corti, Copenhagen, Denmark. His research interests include speech recognition and representation learning for speech.

Jakob D. Havtorn received the bachelor's degree in physics and master's degree in mathematical modeling and computation from the Technical University of Denmark (DTU), Copenhagen, Denmark, in 2016 and 2018, respectively. He is currently working toward the industrial Ph.D. degree with Corti and DTU, supervised by Associate Professor Jes Frellsen, an Adjunct Associate Professor Lars Maaløe and Professors Søren Hauberg and Ole Winther. His research interests include deep generative latent variable models, uncertainty quantification and representation learning for speech.

Joakim Edin received the master's degree in biomedical engineering from the Technical University of Denmark, Copenhagen, Denmark, in 2019. He is currently working toward the Ph.D. degree with the University of Copenhagen, Copenhagen. He was an Industrial Machine Learning Researcher with the Danish company Corti, Copenhagen, Denmark. He is supervised by Professor Søren Brunak and an Adjunct Associate Professor Lars Maaløe. His research focuses on representation learning from medical conversations and text.

Christian Igel received the Diploma degree in computer science from the Technical University Dortmund, Dortmund, Germany, in 1997, the Dr.rer.nat. degree from Bielefeld University, Bielefeld, Germany, in 2002, and the Habilitation degree from Ruhr-University Bochum, Bochum, Germany, in 2010. From 2003 to 2010, he was a Junior Professor with the Institute for Neural Computation, Ruhr-University Bochum. He joined DIKU, the Department of Computer Science, University of Copenhagen (UCPH), Copenhagen, Denmark, in 2010. He is currently a Full Professor with DIKU and the Director of the SCIENCE AI Centre with UCPH. He is a Fellow of the European Lab for Learning and Intelligent Systems.

Katrin Kirchhoff is currently the Director of Applied Science with Amazon Web Services, where she heads several teams in speech and audio processing. She was a Research Professor with the UW, Seattle, WA, USA, for 17 years, where she co-founded the Signal, Speech and Language Interpretation Lab. She was on the editorial boards of *Speech Communication and Computer, Speech, and Language*, and was a Member of IEEE SPEECH TECHNICAL COMMITTEE.

Shang-Wen Li received the Ph.D. degree from the Spoken Language Systems Group, MIT CSAIL, Cambridge, MA, USA, in 2016. He was with Apple Siri, Amazon Alexa and AWS. He is currently the Research and Engineering Manager with Meta AI. He co-organized the workshop of *Self-Supervised Learning for Speech and Audio Processing* at NeurIPS (2020) and AAAI (2022). His research focuses on self-supervised learning in speech and its application to language understanding.

Karen Livescu (Senior Member, IEEE) received the Ph.D. degree from the Spoken Language Systems Group, MIT, Cambridge, MA, USA. She is currently a Professor with TTI-Chicago, Chicago, IL, USA. She is also an ISCA Fellow and an IEEE Distinguished Lecturer, and has been the Program Chair for ICLR, Interspeech, and ASRU. She works on a variety of topics in speech and language processing and machine learning. She has co-organized multiple workshops on machine learning for speech processing, multi-view representation learning, and self-supervised representation learning.

Lars Maaløe received the Ph.D. degree from the Technical University of Denmark (DTU), Copenhagen, Denmark, in 2018. He was nominated Ph.D. of the year by the Department for Applied Mathematics and Computer Science. In the past, he was with financial sector and various tech companies such as Apple. He is a co-founder and the CTO of Corti, Copenhagen, Denmark, a voice-based digital assistant powered by artificial intelligence. He also holds an appointment as Adjunct Associate Professor with DTU.

Tara N. Sainath received the Ph.D. degree from Spoken Language Systems Group, MIT, Cambridge, MA, USA. She is currently the Principal Research Scientist with Google, Mountain View, CA, USA. Her research interests include applications of deep neural networks for automatic speech recognition, and has been very active in the community organizing workshops and special sessions on this topic. She is an IEEE and ISCA Fellow. She was the recipient of the 2021 IEEE SPS Industrial Innovation Award.

Shinji Watanabe (Senior Member, IEEE) is currently an Associate Professor with CMU, Pittsburgh, PA, USA. He was a Research Scientist with NTT, Tokyo, Japan, Visiting Scholar with Georgia Tech, Atlanta, GA, USA, Senior Principal Research Scientist with MERL, Cambridge, MA, USA, Associate Research Professor with JHU, Baltimore, MD, USA. He has authored or coauthored more than 300 peer-reviewed papers. He is a Senior Area Editor of IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING. He was/has been a Member of several technical committees, including the APSIPA SLA, IEEE SPS SLTC, and MLSP.