International Journal of Speech Technology (2024) 27:765–779
https://doi.org/10.1007/s10772-024-10121-9

Efficiency-oriented approaches for self-supervised speech representation learning

Luis Lugo[1] · **Valentin Vielzeuf**[1]

Received: 22 January 2024 / Accepted: 21 June 2024 / Published online: 19 August 2024

Abstract
Self-supervised learning enables the training of large neural models without the need for large, labeled datasets. It has been generating breakthroughs in several fields, including computer vision, natural language processing, biology, and speech. In particular, the state-of-the-art in several speech processing applications, such as automatic speech recognition or speaker identification, are models where the latent representation is learned using self-supervised approaches. Several configurations exist in self-supervised learning for speech, including contrastive, predictive, and multilingual approaches. There is, however, a crucial limitation in the majority of existing approaches: their high computational costs. These costs limit the deployment of models, the size of the training dataset, and the number of research groups that can afford research with large self-supervised models. Likewise, we should consider the environmental costs that high energy consumption implies. Efforts in this direction comprise optimization of existing models, neural architecture efficiency, improvements in finetuning for speech processing tasks, and data efficiency. But despite current efforts, more work could be done to address high computational costs in self-supervised representation learning.

Keywords Self-supervised learning · Speech representation · Knowledge distillation · Transfer learning

1 Introduction

Labeling data for speech recognition models is expensive, a limitation that hinders research and industrial applications in speech processing (Hsu et al., 2021; Mohamed et al., 2022). Moreover, low-resource languages and dialects that have no written resources create limitations for supervised training of deep learning architectures. These limitations motivate the need for self-supervised models to learn latent speech representations (Hsu et al., 2021; Zhang et al., 2020).

Self-supervised approaches for speech representation learning can be based on consistency or self-training (Zhang et al., 2020). In self-training, a teacher model generates labels to train a student model. The self-training process is iterative, using several iterations to get the final model. For instance, CombinedSSL (Combined semi-supervised learning) uses four iterations to train a state-of-the-art speech representation model (Zhang et al., 2020). To train models in a semi-supervised way, methods combine labeled and unlabeled data in the training dataset. In such cases, methods use a supervised loss for the labeled data, combining it with an unsupervised objective or pseudo labels for the unlabeled data (Huang et al., 2022). Noisy student from computer vision is an example (Xie et al., 2020).

On the other hand, consistency defines a task and trains the model on unlabeled data to learn a latent representation (Zhang et al., 2020). For example, bootstrap your own latent (BYOL) trains a model comprising online and target networks, which interact and learn from each other using augmented samples of input data (Grill et al., 2020). Once models learn a latent representation, they can be fine-tuned for a downstream task—automatic speech recognition or speaker identification, for example—in a supervised way (Chen et al., 2020; Huang et al., 2022; Zhang et al., 2020).

Currently, self-supervised models generate state-of-the-art results when learning latent representations, and their results in speech processing are competitive (Parcollet et al., 2023). Downstream tasks in speech processing strongly benefit from latent representations (Mohamed et al., 2022).

✉ Luis Lugo
luiseduardo.1ugomartinez@orange.com

Valentin Vielzeuf
valentin.vielzeuf@orange.com

[1] Orange, 4 Rue du Clos Courtel, Cesson-Sevigne, 35510 Brittany, France

Those tasks include phoneme recognition (PR), keyword spotting (KS), intent classification (IC), speaker identification (SID), emotion recognition (ER), automatic speech recognition (ASR), query by example spoken term detection (QbE), slot filling (SF), automatic speaker verification (ASV), and speaker diarization (SD) (Chang et al., 2022).

Despite competitive results in speech processing, most existing self-supervised learning (SSL) models require several graphics processing units (GPUs) for days to pretrain their neural architectures. This requirement implies large computational costs, which causes several limitations. First, it hinders the training and deployment of speech models in computing platforms with low resources, such as edge devices. It can also limit the size of the training dataset. Secondly, reproducibility is challenging, as few laboratories can afford large computational costs. Finally, it creates environmental issues because of the high energy consumption during training (Chen et al., 2023; Gaol et al., 2023; Wu et al., 2022). Therefore, large computational costs motivate the need for efficient approaches in speech representation learning.

In this comprehensive review, we structure the sections as follows. Section 2 presents an overview of recent models for self-supervised representation learning. Then, efficiency-related approaches are organized into the optimization of existing models (Sect. 3), architecture efficiency (Sect. 4), fine-tuning efficiency (Sect. 5), and data efficiency (Sect. 6). Finally, Sect. 7 presents conclusions and future directions.

2 Self-supervised learning for speech

Most self-supervised learning architectures for speech processing use a combination of convolutional and self-attention layers. Overall, convolutional layers are good at getting local features. They are widely used in vision because they can model edges and shapes by extracting positional information from local features. For global modeling, however, convolutional layers are limited. That explains why convolutional networks need a large stack of layers to grasp global features. Transformers, on the other hand, are good at modeling global context. They are also easier to train in parallel (Gulati et al., 2020).

Recent work has been done with transformer architectures to deal with local feature limitations by adding positional information (Wu et al., 2020; Yang et al., 2019; Yu et al., 2018). Another alternative is to use a transformer channel and a convolutional channel, concatenating at the end the output of the two channels (Bello et al., 2019). For speech processing, a neural configuration known as a Convolution-augmented Transformer, or Conformer (Gulati et al., 2020), organically combines convolutional layers and self-attention layers from transformers, using feed-forward layers in a macaron configuration to wrap intermediate layers. Also, multiple heads for attention enable the model to focus on different parts of the input sequences.

In particular, two approaches have emerged from recent self-supervised speech representation models: Hidden unit BERT (HuBERT) (Hsu et al., 2021) and wav2vec2 (Baevski et al., 2020). The latter emerged because of its solid performance in speech representation and downstream tasks, paired with the availability of pretrained multilingual models. The former emerged because of its simplicity and stability, as well as the way it trains the neural model, which is similar to classical frame level ASR, spearheading further work from research extensions (Mohamed et al., 2022).

Both HuBERT and wav2vec2 combine convolutional and self-attention layers in a dual channel configuration, where a teacher channel generates latent representations, and a student channel learns to replicate the latent space during pretraining. For the pretraining loss, HuBERT uses a predictive loss, while wav2vec2 uses a contrastive loss. Therefore, based on the pretraining loss, we can group self-supervised speech representation approaches into predictive and contrastive methods (Fig. 1).

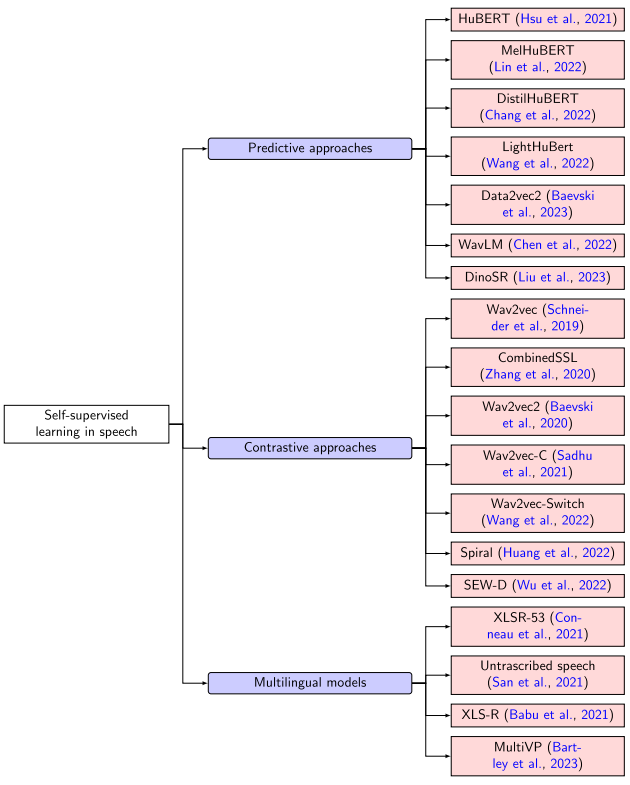

Fig. 1 Recent self-supervised approaches for speech representation learning, including contrastive and predictive approaches, and multilingual models

2.1 Predictive self-supervised approaches

Self-supervised models for speech representations face three challenges. First, input utterances have variable numbers of sound inputs. Secondly, there is no lexicon for input sounds. Thirdly, input sounds are not clearly segmented in the input utterances, as speech data is variable (Hsu et al., 2021). Several models have been recently proposed to address these challenges (Fig. 1), using a predictive loss during pretraining.

HuBERT (Hsu et al., 2021) uses a predictive loss for self-supervised representation learning. It also uses masked segments of input data for pretraining, similar to the training of language models with masked words in large natural language datasets (Devlin et al., 2019). Formally, the loss is calculated as follows (Hsu et al., 2021):

$$X = [x_1, \cdots x_T] \tag{1}$$

$$\tilde{X} = r(X, M) \tag{2}$$

$$z_t = h(x_t) \tag{3}$$

$$\mathcal{L} = \sum_{t \in M} \log p_f(z_t | \tilde{X}, t) \tag{4}$$

where X is an input utterance of length T, M is the set of indices to be masked, r replaces x_t with a mask embedding $\tilde{x}$ if $t \in M$, h is a clustering model, and f is a masked prediction model that predicts a distribution p_f over the target indices at each timestep.

By learning randomly masked sounds from the input utterances, HuBERT learns both the speech representation and the language model, which is closely related to the long-term relationships between inputs, as such long-term relationships come from the language itself. The loss is calculated only on the masked inputs so that the model learns the two features at the same time (Hsu et al., 2021).

The neural architecture of HuBERT comprises a convolutional module, a self-attention module, and a projection module. For the self-supervised training, labels come from the cluster assignments that kmeans generates. The model should learn to match the cluster assignments from kmeans during pretraining. Overall, the training involves three stages. The first stage trains the model with kmeans assignments from Mel-frequency cepstral coefficients (MFCC). The second stage trains the model with kmeans assignments from the representation learned in the first stage. The third stage performs fine-tuning with ASR as the downstream speech application, replacing the projection module with a softmax layer for the connectionist temporal classification (CTC) loss, and freezing the convolutional part of the architecture. Metrics for the self-supervised training iterations include phoneme purity, entropy, and phoneme normalized mutual information. During the supervised fine-tuning, HuBERT uses the word error rate (Hsu et al., 2021).

WavLM (Chen et al., 2022) is an approach very similar to HuBERT. It emphasizes spoken content modeling and the preservation of speaker identity. Most methods use utterances containing speech from a single speaker for pretraining. In contrast, WavLM combines utterances from different speakers to perform data augmentation of the input sequence. it also extends the self-attention mechanism with gated relative position bias to improve the quality of the learned representations (Chen et al., 2022; Mohamed et al., 2022).

DinoSR (Self-distillation and online clustering for self-supervised representation learning) replaces the kmeans in HuBERT with an online clustering approach (Liu et al., 2023). It combines masked input pretraining and online clustering for target discretization. Similar to other self-training approaches, the teacher weights are updated using the exponential moving average, while the student uses backpropagation to adjust the weights during training. There is no need for approximation in targets—like wav2vec2 or similar methods—because the online clustering targets for the teacher do not need to be backpropagated. DinoSR results show improvements in ASR against base models of previous methods (Baevski et al., 2020; Hsu et al., 2021), as well as improvements in results for SUPERB (Speech processing universal performance benchmark) (Yang et al., 2021).

Data2vec2 (Baevski et al., 2023) is a teacher—student model trained with a predictive loss. The teacher weights are updated from the student weights using an exponential moving average for every update. The proposed model supports a multimodal input, using the same loss for image, text, or speech. Feature extractors change to adapt the model to the input data, but the core transformer module remains the same.

To improve the training efficiency of Data2vec2, the training relies on contextualized training, using the whole sample to provide context to the model. Training also reuses the target representation from the teacher for every masked version of the student input. The neural architecture has a fast convolutional decoder and efficient data encoding. The predictive loss is calculated from the L2 distance between the student output for the masked input and the target output the teacher generates for the unmasked input sample (Baevski et al., 2023).

Data2vec2 can be up to ten times faster than existing self-learning representation models in speech, but it still manages to have a better performance. The base model is pre-trained using LibriSpeech (Panayotov et al., 2015), while the large one uses LibriLight (Kahn et al., 2020), fine-tuning for ASR and using a 4-gram language model during decoding.

2.2 Contrastive self-supervised approaches

Contrastive approaches replace the predictive loss during self-training for a contrastive loss. A contrastive loss in a teacher—student configuration is defined as follows (Chen et al., 2020; Huang et al., 2022):

$$\phi(a,b) = \frac{a^T b}{\|a\|\|b\|} \tag{5}$$

$$\mathcal{L} = -\sum_{i=1}^{N} \log \frac{e^{\phi(z_i,z_i')/\tau}}{\sum_{j\in D_i} e^{\phi(z_i,z_j')/\tau}} \tag{6}$$

where z is the latent representation from the student network, z' is the latent representation from the teacher network, ϕ is the cosine distance between vectors a and b, τ represents a temperature parameter, and D_i is the set of distractors for the z_i representation.

Wav2vec2 (Baevski et al., 2020) is quite a popular approach in the contrastive loss category. It combines masked input timesteps and quantization through Gumbel SoftMax (Jang et al., 2016) to learn latent speech representations. Follow-up studies of wav2vec2 include wav2vec-C (Sadhu et al., 2021), which extends the loss of wav2vec2 with a consistency term to reconstruct the input from quantized representations, and wav2vec-switch (Wang et al., 2022), which adds noise robustness to the contextualized representation learning of wav2vec2.

Self-supervised perturbation invariant representation learning (SPIRAL) also pretrains the model using a contrastive loss (Huang et al., 2022). SPIRAL comprises a subsampling convolutional module, a self-attention module, a second subsampling convolutional module, and a second self-attention module. Then, a projection module generates the architecture output. An additional predictor module for the student only improves performance. Both teacher and student have the same architecture, except for the student predictor module. SPIRAL gets results similar to or better than wav2vec2. It, however, uses no quantization and considerably reduces training costs.

The input to the student part of the model is perturbed with noise, and the student's objective is to learn a denoising representation as close as possible to the teacher output, considering that the teacher input has no noise added to it. The utterance representation uses log Mel filter banks with 128 filters. In addition to SpecAugment and dropout (Park et al., 2019, 2020), the input is also regularized by noise addition during pretraining, using a method known as multi-condition pretraining (Chiba et al., 2019; Seltzer et al., 2013), where noise from a dataset is added to the input utterance avec varying degrees of signal to noise ratio. To avoid learning trivial weights, SPIRAL uses two methods. First, it randomly adds padding to the teacher's input, avoiding any positional information becoming the student's learning objective. Secondly, it uses an in-utterance contrastive loss, mixing the negative samples in the learning objective (Huang et al., 2022).

CombinedSSL improves the state-of-the-art in ASR by combining a conformer architecture, wav2vec2 pretraining, and noisy student self-training, where SpecAugment provides the noise for the input speech sequences (Zhang et al., 2020). CombinedSSL uses a convolutional subsampling module, a linear module, and a conformer block. The convolutional subsampling module can be seen as a feature encoder, while the conformer module serves as a context network. The positional embeddings of the self-attention are removed with no negative impact on the final results. Removing positional embeddings has no negative impact because the convolutional module learns relative positional information, according to previous research (Wang et al., 2020).

For CombinedSSL pretraining, the original raw waveforms in wav2vec2 are replaced by 80 Mel log spectral coefficients, which makes the input sequence shorter. Another variation against wav2vec2 is the removal of the quantization module, using a linear layer instead. CombinedSSL uses 512 TPUs for the largest model during four days. LibriSpeech is the dataset to test supervised ASR, while Librilight provides unlabeled data for petraining.

2.3 Multilingual models

Multilingual approaches are particularly important for languages and dialects without written resources for training, as cross-lingual transfer facilitates learning speech representations in low-resource settings (Abdullah et al., 2023; San et al., 2021). Additionally, cross-lingual pretraining outperforms monolingual pretraining after fine-tuning for speech processing tasks, particularly in low resource languages (Conneau et al., 2021).

The wav2vec2 architecture serves as the base for several multilingual approaches, including XLS-R (Babu et al., 2021) and XLSR-53 (Conneau et al., 2021). XLSR-53 pretrains a large architecture with unlabeled speech data in multiple languages. It uses a shared quantization module, which forces the model to share discrete tokens across languages, learning bridges between different languages. The model supports 53 languages. XLS-R extends the training of XLSR-53 going from 56 thousand hours to nearly half a million hours of publicly available speech data. Language coverage also improves, going from 53 languages to 128 languages, and evaluation is performed with various speech processing tasks, including language identification, ASR, and speech translation.

MultiVP (Bartley et al., 2023) pretrains a conformer-based model with a multilingual dataset, fine-tuning it to perform spoken language identification. Lower layers of the model encode enough information to discriminate languages and identify speech sounds. The fine-tuned model gets similar results to XLSR in language identification, though it uses only around 20% of parameters. MultiVP can also identify unseen languages, showing its ability to realize zero-shot language identification.

2.4 Limitations of self-supervised models

The word error rate (WER) is widely used in evaluating speech representation models after supervised fine-tuning for ASR (Table 1). The error rate is the relationship between correct input–output words and the testing set size (Graves, 2012):

$$WER = \frac{1}{|S'|} \sum_{(x,z)\in S'} \begin{cases} 0 \text{ if } h(x) = z \\ 1 \text{ otherwise} \end{cases} \tag{7}$$

where S' is the test set, x is the input, z is the expected output, and h is the learning model.

But alternative metrics have been proposed to improve the evaluation of speech representations, including *ABX* and D_{SSIMI} (Chung & Glass, 2018; Dunbar et al., 2021; Kahn et al., 2020; Nguyen et al., 2020). D_{SSIMI} calculates the distance between a pair of words from a semantic perspective, correlating the distance of learned representations with a human judgment of semantic similarity. On the other hand, *ABX* is an acoustic—phonetic metric. It measures how well separated phonetic categories are in a latent representation space (Nguyen et al., 2020). For a given pair of sounds, we calculate the probability that two sounds of the same category are closer to one another than two sounds of different categories. For a triphone (a, b, x), from categories A and B, we want to know the probability of

$$d(a, x) < d(b, x) \tag{8}$$

$$a \in A, b \in B, x \neq a \in A \tag{9}$$

where d is a framewise distance. It could be the Kullback–Leibler or the angular distance (Nguyen et al., 2020).

Though the evaluation of speech representation models is mostly done with ASR, some datasets have been proposed to test representation models with other speech processing tasks. Those datasets include SUPERB and LeBenchmark (Evain et al., 2021; Yang et al., 2021). SUPERB includes phoneme recognition, keyword spotting, intent classification, speaker identification, emotion recognition, automatic speech recognition, query by example spoken term detection, slot filling, automatic speaker verification, and speaker diarization (Chang et al., 2022).

Additionally, few existing models use half-precision floating point numbers, even though this technique can half the memory requirements and accelerate the arithmetic computations on recent GPUs (Micikevicius et al., 2018). A similar issue happens with dynamic batching (Tyagi & Sharma, 2020), a procedure that avoids wasting computing resources on the padded portion of speech mini-batches.

Despite the groundbreaking results in speech processing, self-supervised models are computationally expensive. Large training costs represent a challenge. Indeed,

Table 1 Recent models for self-supervised speech representation learning

Model	WER (%)	GPUs	Size (M)	Hours
Transformer Acoustic (Wang et al., 2020)	4.8	32	93	96
BLSTM Acoustic (Wang et al., 2020)	6.6	32	163	96
Hubert Large (Hsu et al., 2021)	3.3	128	300	99.75
Hubert XLarge (Hsu et al., 2021)	2.9	256	1000	99.75
wav2vec2 Base (Baevski et al., 2020)	4.8	64	95	38.4
wav2vec2 Large (Baevski et al., 2020)	3.3	128	317	180
CombinedSSL XXL (Zhang et al., 2020)	2.7	512*	1000	168
CombinedSSL XXL+ (Zhang et al., 2020)	2.6	512*	1050	168
Data2vec2 Base (Baevski et al., 2022)	5.2	16	–	43.3
Data2vec2 Large (Baevski et al., 2023)	3.5	64	–	76.7
Spiral Base (Huang et al., 2022)	6.1	16	95	31.2
Spiral Large (Huang et al., 2022)	3.5	32	317	174
DinoSR (Liu et al., 2023)	6.7	16	–	180

Recent models for self-supervised speech representation learning. Results are for the LibriSpeech test-other dataset, using WER as the evaluation metric, and ASR as the downstream task for fine tuning the neural architectures.

*This model uses TPUs

there is a trade-off between self-training computational costs and speech representation performance (Fig. 2). For example, the best-performing method, CombinedSSL (Zhang et al., 2020), uses 512 Tensor Processing Units (TPUs) for four days to train its XXL+ version. In contrast, BLSTM (Wang et al., 2020) requires only 32 GPUs for 96 h, but its representation performance degrades, passing from a WER of 2.6% to 6.63% for the LibriSpeech test-other dataset (Table 1).

The computational cost of large self-supervised models hinders the study of other training recipes and the reproduction of experimental results, as few researchers can afford the cost (Lin et al., 2022). Computational costs limit the number of laboratories with enough computational resources to perform research, which creates a barrier to developing new approaches (Parcollet et al., 2023; Wang et al., 2023). Computational costs also have environmental implications, as training requires considerable amounts of energy (Parcollet et al., 2023; Wu et al., 2022).

Current efforts to deal with computational costs in large self-supervised models include data efficiency (Maekawa et al., 2023; Reed et al., 2022; Wang et al., 2018), fine-tuning with parameter-efficient transfer learning (Hu et al., 2021; Zhang et al., 2023), optimization of neural architecture components (Dao et al., 2022; Lee-Thorp et al., 2022; Moumen & Parcollet, 2023; Wu et al., 2020), and optimization of existing speech models (Chang et al., 2022; Lin et al., 2022).

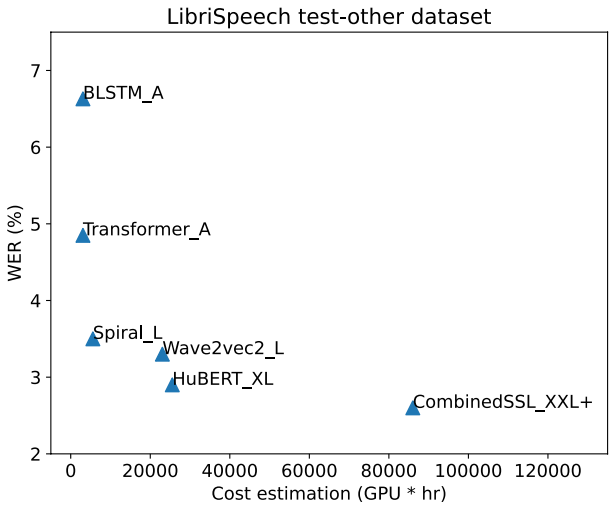

Fig. 2 Existing models for self-supervised speech representation learning. This trend illustrates the trade-off between computational costs and representation performance

3 Towards optimization of existing speech models

As mentioned, self-supervised models for speech representation have generated competitive results in multiple downstream speech processing tasks. However, those models are quite large, requiring a lot of computational resources for training. One alternative to reduce model size is knowledge distillation (Allen-Zhu & Li, 2020), where a small student model learns from a large teacher model, which has been pretrained and remains frozen during student training (Peng et al., 2023).

Drawing inspiration from DistilBERT (Sanh et al., 2019), DistilHuBERT (Chang et al., 2022) is a knowledge distillation approach that reduces the model size by 75% and speeds up inference by 73%. DistilHuBERT comprises a convolutional module with seven layers, two self-attention layers, and three multitask layers, which learn to predict layers 4, 8, and 12 from the transformer in HuBERT. These layers were chosen because close layers tend to encode similar information. Layers in DistilHuBERT are initialized by their counterparts in the teacher module. Training requires a single GPU for 200k iterations, which takes roughly 55 h, with a model size of 24 M parameters, compared to the 95 M from HuBERT. For the loss, DistilHuBERT optimizes both the L1 distance and the cosine similarity, using a lambda parameter as a factor of the cosine similarity.

For the speech processing tasks, a weighted sum of the multitask layers generates the output of the DistilHuBERT architecture. Results are comparable to those of the teacher model, demonstrating the ability of the small student model to capture the HuBERT speech representation while pretraining with the LibriSpeech dataset and testing with SUPERB (Chang et al., 2022).

LightHuBERT (Wang et al., 2022) also aims at optimizing the HuBERT architecture. It relies on knowledge distillation to learn a once-for-all transformer model. The teacher is a HuBERT base model, while the student learns by predicting masked inputs in an iterative process. The transformer in LightHuBERT comprises subnets with sharable weights and several configuration parameters, enabling a random search to adjust the model to different resource constraints. Results outperform the teacher model and improve DistilHuBERT performance, while the number of parameters is reduced by up to 29% against the base model.

Similarly, FitHuBERT (Lee et al., 2022) proposes a thinner and deeper neural configuration than HuBERT, without reducing the number of layers. Training is performed with knowledge distillation from a pretrained HuBERT model, reducing the model size and inference time.

The student architecture in knowledge distillation methods is manually designed. And it does not change

during training. However, modifying student architectures can have a considerable impact on model results, even for student architectures with similar sizes (Ashihara et al., 2022). Therefore, a joined distillation and pruning approach for speech SSL has been recently proposed, using HuBERT (DPHuBERT) or WavLM (DPWavLM) as the teacher models (Peng et al., 2023). The pruning performed during training is structured, which enables the removal of groups of parameters—an entire layer, for instance—from the base model (Peng et al., 2023). In contrast, unstructured pruning sets to zero individual weights in the model, generating a sparse architecture without speedup. The joined approach improves over previous knowledge distillation approaches in most tasks of the SUPERB dataset (Peng et al., 2023).

Another approach to reduce the cost of self-supervised learning is MelHuBERT (Lin et al., 2022). Contrary to DistilHuBERT, DPHuBERT, or DPWavLM, MelHuBERT does not use knowledge distillation. Instead, it is a simplified version of HuBERT that has twelve self-attention layers and a weighted sum of all the layers for downstream tasks, as each self-attention layer best encodes a certain aspect of the input speech sequence (Pasad et al., 2021, 2023). For example, the sixth layer best represents phoneme information and cluster purity. Though results degrade from the original HuBERT model in terms of phoneme recognition and speaker identification, they remain better than other existing models such as wav2vec (Schneider et al., 2019), vq-wav2vec (Baevski et al., 2020), and multilayer LSTMs (Yeh & Tang, 2022).

MelHuBERT reduces the multiplication and addition calculations (MAC) by 33%. As the input is a 40-dimensional Mel log spectrogram, input sequences are shorter. Additionally, the spectrogram calculation removes the need for the convolutional module for feature extraction, further reducing the model size. MelHuBERT also reduces the iterative training scheme, relying only on two training stages. Training of the model requires a single GPU during 150hrs, using a subset of LibriSpeech (Lin et al., 2022).

It is also possible to train HuBERT in an academic setup (Chen et al., 2023), using only 8 GPUs to train the model. This approach gets similar results to the original model. Optimizations include mixed precision training and gradient accumulation to match the batch size of the original model, keeping in mind that the batch size has a considerable impact on model results.

Along with optimizations to modify the HuBERT architecture, there are also efforts to optimize the wav2vec architecture. Proposed approaches optimizing wav2vec include squeezed and efficient wav2vec2 with disentangled attention (SEW-D) (Wu et al., 2022), stochastic squeezed and efficient wav2vec2 (S-SEW) (Vyas et al., 2022), and PARP (Prune Adjust and Re-Prune) (Lai et al., 2021).

PARP (Lai et al., 2021) prunes a pretrained wav2vec2 model to improve its computational complexity, obtaining a sparsity mask and updating it while finetuning the speech representation for downstream tasks.

In contrast, SEW-D (Wu et al., 2022) proposes four changes for optimizing wave2vec2, reducing inference and pretraining time: a squeezed context module, a compact feature extractor, MLP projection heads, and a disentangled attention instead of multiheaded attention. Disentangled attention uses relative positional embeddings and context embeddings separately, combining them in the weight calculation with content to content, content to positional, and positional to content attention. This disentangled configuration outperforms multiheaded attention in transformer models.

Instead of a linear projection layer, SEW-D uses a projection MLP with two linear layers, each one with ReLU activations and batch normalization. The compact feature extractor decreases the number of convolutional filters in the wav2vec2 original architecture, reducing the size of the first convolutional layer and increasing subsequent layers' size as the frame rate decreases. This convolutional configuration helps to more evenly distribute the computations in the forward and backward pass. After the compact feature extractor, the squeezed context module takes as input a downsampled tensor, processes it, and upsamples the output before calculating the pretraining loss. This squeeze reduces the computations in the transformer module, improving overall performance (Wu et al., 2022).

The base configurations of SEW-D are pretrained for 100k iterations and finetuned for 80k iterations. Pretraining takes a day with 8 GPUs. For comparison against wav2vec2 ASR results, pretraining goes up to 400k iterations. Optimizations allow SEW-D to get between 2 to 3 times faster inference against wav2vec2 baseline configurations. It also has faster pretraining times, using 8 GPUs for 24 h in base configurations and getting comparable or better word error rates (Wu et al., 2022).

Though SEW-D reduces wave2vec2 complexity, the performance degrades. Thus, S-SEW (Vyas et al., 2022) proposes a probabilistic approach to smooth the trade-off between performance and computational complexity. To do so, the squeezing factor in SEW is replaced with a stochastic factor, choosing it randomly for each iteration from a uniform distribution. S-SEW also performs a stochastic pooling of key and value in the self-attention layers, which further improves computational efficiency.

4 Towards neural architecture efficiency

In general, self-attention layers have contributed to breakthrough results in modeling speech, language, vision, biology, and other domains (Mohamed et al., 2022; Poli et al.,

2023). Self-attention layers can generalize to unseen data and tasks because of their capacity for in-context learning. They can also learn dependencies from the whole input without any restrictions in context. Self-attention scales quadratically to sequence length, becoming increasingly expensive as the input length grows. Recent work shows they use only a small portion of their quadratic capabilities when processing language (Poli et al., 2023).

Sub-quadratic alternatives for self-attention layers in transformers have been proposed in recent work (Fig. 3). These sub quadratic layers use linear (Schlag et al., 2021; Zhai et al., 2021), learnable (Kitaev et al., 2020; Wang et al., 2021), sparse (Child et al., 2019; Roy et al., 2021; Tu et al., 2022), or low-rank (Choromanski et al., 2021; Wang et al., 2020) approximations to accelerate layer calculations (Poli et al., 2023; Tay et al., 2022). Though these approximations improve the efficiency of self-attention layers by helping alleviate their quadratic relationship to the input sequence length, they reduce the quality of the models (Dao et al., 2022), requiring a hybrid combination with standard self-attention layers to get a similar quality (Dao et al., 2022; Mehta et al., 2022; Poli et al., 2023).

But some architectures scale subquadratically to sequence length without compromising model quality. They include the Sumformer (Parcollet et al., 2023), the Fnet (Lee-Thorp et al., 2022), and the Hyena hierarchy (Poli et al., 2023).

The Hyena hierarchy (Poli et al., 2023) is an architecture that implements a recurrence of two subquadratic primitives: a long convolution and an element-wise multiplication. The number of recurrences sets the size of the operator. Results match transformer models, but computational costs are lower and the model is faster—especially at long sequences. This improvement happens because of its sub-quadratic scaling in terms of sequence length.

The Fnet (Lee-Thorp et al., 2022) replaces self-attention completely. Instead, it uses the Fast Fourier transform (FFT) plus a feed-forward layer. This change speeds up the computations because FFTs are fast to calculate in GPUs. It also reduces the number of network parameters because the learnable matrices in self-attention layers are no longer necessary. The efficiency boost from this change can go up to 97% against models using self-attention layers, while the degradation in modeling results remains small.

Similar to Fnets, Sumformers (Parcollet et al., 2023) replace the self-attention layers altogether. Weights in self-attention layers of some speech processing architectures behave close to a mean function, suggesting self-attention is not essential for speech recognition architectures processing acoustic utterances. Instead of using multiheaded self-attention, it is possible to replace it with a linear alternative, which calculates the mean vector of all time steps in a speech utterance. Time-specific information complements the mean vector, generating an approach known as summary

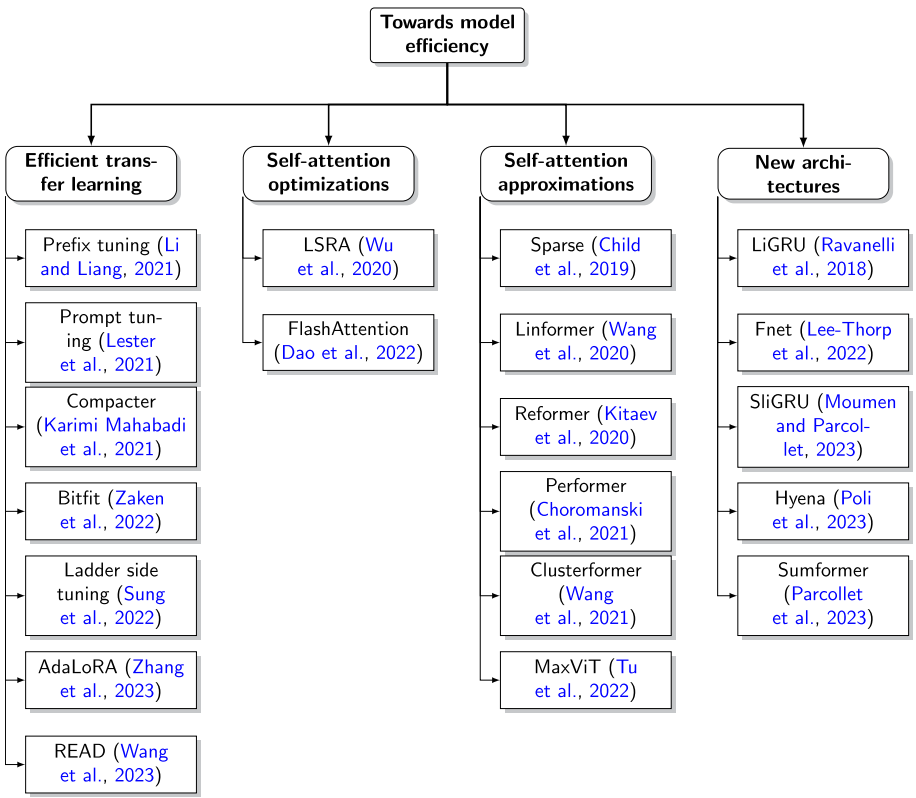

Fig. 3 Methods proposing efficiency-oriented approaches that can deal with the computational costs of self-supervised learning architectures, including new architectures, self-attention improvements, and efficient transfer learning

mixing. Using summary mixing instead of multiheaded self-attention, the sumformer processes speech data faster than existing speech processing approaches without degradation in modeling performance. The sumformer generates up to 27% improvement in training and inference time. It also reduces in half the RAM requirements.

Other methods optimize self-attention layers with long short range attention in a lite transformer (Wu et al., 2020), or GPU efficient operations in FlashAttention (Dao et al., 2022), without using approximations.

FlashAttention (Dao et al., 2022) improves self-attention layers' efficiency by optimizing the input—output (IO) memory operations in the GPU. Overall, GPUs have two kinds of memories. A small SRAM is associated with each kernel, while a large, slower, high-bandwidth memory is shared between all the kernels. Memory-intensive operations, like the matrix operation of the self-attention layers, have their bottleneck at the read-write RAM access. In contrast, compute-intensive operations have their bottleneck in the number of arithmetic operations that must be realized. As self-attention is primarily a memory-intensive operation, FlashAttention reduces the number of IO operations by tiling, assigning a matrix operation to a single kernel, and saving some results from the forward pass to share in the subsequent backward pass.

The optimization of IO operations can speed up models using FlashAttention by three times. The impact of the optimization is higher in long sequences. It is also possible to insert input approximations, such as sparsity, to combine memory optimization and input length reduction(Dao et al., 2022).

The lite transformer (Wu et al., 2020) proposes a Long Short Range Attention (LSRA), which preserves the capacity of self-attention layers to model relationships in the input sequence. LSRA removes the dimensionality reduction of the bottleneck configuration observed in the feed-forward modules surrounding the self-attention layers in the transformer's original configuration. Then, it replaces the self-attention layers for a dual-channel configuration. The first channel comprises self-attention heads for long-range modeling. The second channel is a convolutional module for short-range modeling. The two channels do the processing in parallel, each with a chunk of the input. Then, the results are concatenated with a feed-forward module.

The number of parameters in LSRA remains the same, but there is no context loss like the loss that the dimensionality reduction produces in the original transformer architecture. No heads in the self-attention module are reduced, while the context reduction is removed by flattening the input feed-forward module. Similarly, the convolutions in the second channel are faster and less affected by sequence length. Efficiency improvements can go up to 2 times. Moreover, using 8-bit quantization and pruning, the model can be up to 18 times smaller, which is ideal for smartphones and edge devices (Wu et al., 2020).

Additional architectural optimizations include improvements to recurrent layers (Moumen & Parcollet, 2023; Ravanelli et al., 2018). Though transformers are widely used in speech processing, recurrent networks continue to find applications in offline and online speech processing (Chung & Glass, 2018; Le et al., 2019; Moumen & Parcollet, 2023; Ravanelli et al., 2018; Wang et al., 2020; Zhang et al., 2016). For example, the DeepSpeech architecture uses convolutional and recurrent layers for ASR (Amodei et al., 2016); therefore, recurrent networks coexist with transformers in speech processing (Moumen & Parcollet, 2023).

Light gated recurrent units (LiGRU) were proposed for ASR (Ravanelli et al., 2018). They reduce by 30% the training time per epoch when compared against standard GRUs. LiGRUs suppress one gate, the reset gate, reducing the number of parameters in the recurrent unit. LiGRUs also replace the tanh activation function for a ReLU function. However, no implementation was released to the community, hindering its adoption. Existing implementations of LiGRUs use no recurrent optimization, making them slower than optimized LSTM implementations. Another serious limitation is the exploding gradient problem, given the unbound nature of ReLU activations. Even though at first it used batch normalization to deal with exploding gradients, large sequences in speech processing make the problem resurge (Moumen & Parcollet, 2023).

An improved version of LiGRUs was recently proposed (Moumen & Parcollet, 2023): Stabilized LiGRU (SliGRU), which adds layer normalization to the recurrent units to solve the exploding gradients. Besides, an optimized implementation using Pytorch and CUDA makes proposed units up to five times faster than LSTMs, with performance improvements in ASR. SliGRUs are also faster than a version using sine activations for dealing with exploding gradients.

5 Towards finetuning efficiency

Parameter efficient transfer learning (PETL) aims at benefiting from pretrained representations and fine-tuning models to perform downstream tasks (Lialin et al., 2023; Wang et al., 2023). Recent self-supervised approaches use large transformer architectures, which are fine-tuned for specific tasks after a pretraining phase, transferring learning from a pretraining objective to a downstream task. As the size of the transformer models increases, fine-tuning the whole model becomes costly because of the increase in the number of trainable parameters, and for several downstream tasks, there must be one fine-tuned model per task (Wang et al., 2023; Zhang et al., 2023)

To address this limitation, several PETL approaches have been proposed to reduce the computational cost of fine-tuning large transformer models. PETL approaches include reparameterization, additive methods, soft prompts, and partial tuning (Wang et al., 2023).

In partial tuning, only certain layers are trained, or only certain parameters of the model, such as the biases, are trained during the fine-tuning phase (Zaken et al., 2022). Prompting approaches concatenate trainable input vectors to the model input, training them during fine-tuning to modify the transformer output, or alter the whole input embeddings, which represents a learnable prompting scheme (Lester et al., 2021; Li & Liang, 2021). Additive methods insert small neural components into the backbone transformer, training only the components during fine-tuning. Components can be layers inside a transformer or external components in a side-tuning or ladder side-tuning configuration (Sung et al., 2022; Wang et al., 2023; Zhang et al., 2020). Finally, low-rank components alongside self-attention layers are part of reparameterization approaches (Hu et al., 2021; Karimi Mahabadi et al., 2021; Zhang et al., 2023).

Low-Rank Adaptation (LoRA) is a reparameterization approach (Hu et al., 2021). LoRA adds low-rank decomposition matrices to each transformer layer and trains the matrices during fine-tuning, keeping the weights of the pre-trained model frozen. LoRA draws inspiration from the fact that learned transformer models have a low intrinsic dimensionality, suggesting that learning during fine-tuning also happens in a low dimensionality. It is the first approach to add low-rank restrictions for downstream tasks only.

Advantages of LoRA include the same or better results than the fine-tuned baseline, reduction of trainable parameters up to 10 thousand times for large models such as GPT3 (Brown et al., 2020), which has 175B parameters, reduction of GPU RAM usage of up to three times, and no latency during inference. As far as the limitations of LoRA are concerned, low-rank matrices are added to the model to avoid latency; however, this addition limits parallelization when using a batch with different downstream tasks. Likewise, heuristics dictate the way low-rank configuration is selected, motivating the need for more principled ways of setting the configuration. Moreover, LoRA does not prioritize the parameters that have the highest impact on the model, which makes the fine-tuning suboptimal. It also adds low-rank matrices to the self-attention layers only, while low-rank matrices for the feed-forward layers have a higher impact (Hu et al., 2021; Zhang et al., 2023).

As an alternative, Adaptive Low-Rank Adaptation, or AdaLoRA (Zhang et al., 2023), creates low-rank matrices for both self-attention layers and feed-forward layers in the transformer model. AdaLoRA dynamically adapts the low-rank matrices during the fine-tuning process. While using singular value decomposition allows us to identify the most important ranks during fine-tuning, its computational cost is too high. Instead, AdaLoRA uses three low-rank matrices PVQ, where P and Q store the singular dimension vectors, while V stores the singular values. V is diagonal; thus, it can be stored in a vector. P and Q are orthogonal so that low-rank dimensions remain independent from each other. After each gradient descent, an importance metric allows us to determine whether a singular value remains important for the model output. Should a singular value be evaluated as not important, its dimensions are not deleted. Instead, the singular value is replaced by a zero in the V matrix. This pruning allows the singular dimensions to be reactivated later if their importance increases again.

Though the aforementioned PETL approaches reduce the number of parameters trained during fine-tuning, memory consumption remains high. A similar issue happens with energy consumption. Similarly, side-tuning fails to account for intermediate layer results, which are crucial for representation learning, taking only intermediate activations. Ladder side-tuning inserts a transformer component to consider intermediate layer results during fine-tuning, but the inserted transformer requires an additional pretraining phase (Wang et al., 2023).

To address these issues, Recurrent adaptation of large transformers (READ) proposes an additive neural component for fine-tuning (Wang et al., 2023). The neural component comprises a joiner network and a recurrent module without any additional attention mechanisms. The joiner takes as input the output hidden states from all the transformer layers, processing them with a feed-forward network. Then, the recurrent module iteratively processes the joiner output. The final model output is the addition of the recurrent component output and the transformer output.

Tests were performed using the T5 architecture, an encoder—decoder transformer model (Raffel et al., 2020). The dataset used in tests is GLUE (Wang et al., 2018), which includes several NLP downstream tasks such as natural language inference, sentiment classification, paraphrase detection, and linguistic acceptability. Limitations of READ include its behavior in low data configurations. Preliminary tests show READ needs more epochs than existing methods when fine-tuning for small datasets (Wang et al., 2023).

In contrast, the advantages of READ include a 50% reduction of RAM and 86% of GPU energy consumption against fine-tuning with the whole model. Also, the number of additional parameters increases in log-linear form against the transformer's size. Memory usage and energy consumption improvements are also observed when comparing READ against existing PETL methods. Another advantage is multitasking. As the neural component added for fine-tuning is small, it is possible to add multiple components, each one for a task in a multitask setup, without the need to fine-tune separately for each task (Wang et al., 2023).

6 Towards data efficiency for model pretraining

SSL models enable data efficiency when finetuning for downstream speech tasks, as pretrained speech representations can facilitate finetuning with small datasets (Ericsson et al., 2022), motivating research for data-efficient finetuning with very low datasets. But there is also the possibility to improve data efficiency for pretraining SSL models. Recent work in data efficiency for pretraining includes dataset distillation (Maekawa et al., 2023; Wang et al., 2018), trimming of speech sequences (Gaol et al., 2023), contrastive learning for generative models (Shi et al., 2020), and hierarchical pretraining (Reed et al., 2022).

Hierarchical pretraining proposes a multiphase approach. First, the model is trained with a general dataset. Then, the pretraining continues with a domain-specific dataset. In the third phase, the model is pretrained with the target dataset before applying the supervised finetuning with the target dataset. In comparison, generalist pretraining uses a large dataset combining many domains in the input samples, which enables the creation of a repository of models. Specialist pretraining uses a domain-specific dataset to learn the model. In both generalist and specialist pretraining, a final phase of supervised finetuning is realized for calculating model results. But in certain cases, generalist pretraining underperforms models trained end-to-end with domain-specific data. Hierarchical pretraining addresses this issue. It can also improve training convergence, making it up to 80 times faster (Reed et al., 2022).

Generative models using multimodal data focused on the commonality between related samples of different modalities - for example, matching images to their text description. Recent work (Shi et al., 2020) extended the relationship to include non-related samples in the loss calculation. Non-related samples are negative examples in a contrastive learning configuration. By doing so, the multimodal variational autoencoders improve their performance over existing models, optimizing the amount of data necessary for training.

Based on previous results from federated learning for speech, short sequences effectively reduce the computational costs of speech models (Gao et al., 2022). Therefore, a systematic analysis of short sequences enables their evaluation in speech processing tasks, trimming speech sequences to 1, 5, 10, and 15 s. Several downstream tasks are analyzed, performing model training with edge platforms like Raspberry Pi and Nvidia Jetson Xavier. Experimental results show the computational cost reduction that a one-line code change represents, with acceptable results in downstream tasks, especially for downstream applications requiring short speech sequences (Gaol et al., 2023).

Data distillation focuses on data efficiency for training neural models. This method generates synthetic samples from a dataset to train models in a few training steps (Maekawa et al., 2023). Synthetic samples enable a form of dataset compression. For example, one synthetic image per class in the MNIST dataset enables training a model with a fixed network initialization in a few iterations (Wang et al., 2018).

7 Conclusions

In sum, this paper presented an overview of self-supervised representation learning for speech processing, including contrastive, predictive, and multilingual models. Then, we introduced efficiency-oriented approaches like optimization of existing models, neural architecture modifications, finetuning improvements, and data efficiency.

There are, however, several future research directions for optimizing self-supervised representation learning in speech processing. First, despite efforts to improve model efficiency (Chen et al., 2023; Parcollet et al., 2023), more work could be done to reduce the computational costs of self-supervised models. Computational costs create challenges when using these models in mobile devices, for training on very large datasets, and from an energy consumption perspective (Mohamed et al., 2022; Parcollet et al., 2023; Wu et al., 2022).

Secondly, given the competitive performance in several speech tasks, the learned representations seem to encode more features than simply speech. Disentangling the latent vectors could allow other applications. For instance, decomposing the latent representation to model ASR and SID simultaneously (Mohamed et al., 2022; Stafylakis et al., 2022). Thirdly, training datasets for natural language models use way more data than the equivalent models in speech. Natural language models could provide semantic context to representation learning, helping them to learn semantic information from language. Yet it is not clear how to integrate them in speech representation approaches (Mohamed et al., 2022). Finally, other research directions include data efficiency for model pretraining (Maekawa et al., 2023) and capturing semantic information in speech, which has a higher abstraction level than phonetic or lexical modeling (Arora et al., 2022; Lai et al., 2021; Mohamed et al., 2022; Seo et al., 2022).

Author contributions All authors participated in the literature review. All authors have reviewed and approved the final version for publication and maintain accountability for all aspects of the article, including integrity and validity.

Funding Not applicable.

Data availability Not applicable because no dataset was generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare that they have no known competing financial or non-financial interests.

Ethical approval Not applicable.

References

Abdullah, B. M., Shaik, M. M., & Klakow, D. (2023). On the nature of discrete speech representations in multilingual self-supervised models. In *Proceedings of the 5th workshop on research in computational linguistic typology and multilingual.* NLP.

Allen-Zhu, Z., & Li, Y. (2020). Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. arXiv preprint arXiv:2012.09816

Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., … Zhu, Z. (2016). Deep speech 2: End-to-end speech recognition in English and Mandarin. In *International conference on machine learning (ICML).* PMLR.

Arora, S., Dalmia, S., Denisov, P., Chang, X., Ueda, Y., Peng, Y., Zhang, Y., Kumar, S., Ganesan, K., Yan, B., Vu, N., Black, A., & Watanabe, S. (2022). Espnet-slu: Advancing spoken language understanding through ESPnet. In *IEEE international conference on acoustics, speech and signal processing (ICASSP).*

Ashihara, T., Moriya, T., Matsuura, K., & Tanaka, T. (2022). Deep versus wide: An analysis of student architectures for task-agnostic knowledge distillation of self-supervised speech models. In *Interspeech.*

Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., Platen, P., Saraf, Y., Pino, J., Baevski, A., Conneau, A., & Auli, M. (2021). Xls-r: Self-supervised cross-lingual speech representation learning at scale. In *Interspeech.*

Baevski, A., Schneider, S., & Auli, M. (2020). vq-wav2vec: Self-supervised learning of discrete speech representations. In *International conference on learning representations (ICLR).*

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in neural information processing systems (NIPS).*

Baevski, A., Babu, A., Hsu, W.-N., & Auli, M. (2023). Efficient self-supervised learning with contextualized target representations for vision, speech and language. In *International conference on machine learning (ICML).* PMLR.

Baevski, A., Hsu, W.-N., Xu, Q., Babu, A., Gu, J., & Auli, M. (2022). Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International conference on machine learning (ICML).* PMLR.

Bartley, T.M., Jia, F., Puvvada, K.C., Kriman, S., & Ginsburg, B. (2023). Accidental learners: Spoken language identification in multilingual self-supervised models. In *IEEE international conference on acoustics, speech and signal processing (ICASSP).*

Bello, I., Zoph, B., Le, Q., Vaswani, A., & Shlens, J. (2019). Attention augmented convolutional networks. In *Proceedings of the IEEE/CVF international conference on computer vision.*

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D. (2020). Language models are few-shot learners. In *Advances in neural information processing systems (NIPS).*

Chang, H.-J., Yang, S.-w., & Lee, H.-y. (2022). Distilhubert: Speech representation learning by layer-wise distillation of hidden-unit Bert. In *IEEE international conference on acoustics, speech and signal processing (ICASSP).*

Chen, W., Chang, X., Peng, Y., Ni, Z., Maiti, S., & Watanabe, S. (2023). Reducing barriers to self-supervised learning: Hubert pre-training with academic compute. In *Interspeech.*

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *International conference on machine learning (ICML).* PMLR.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X., Wu, J., Zhou, L., Ren, S., Qian, Y., Qian, Y., Wu, J., Zeng, M., Yu, X., Wei, F. (2022). Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing, 16*(6), 1505–1518.

Chiba, Y., Nose, T., & Ito, A. (2019). Multi-condition training for noise-robust speech emotion recognition. *Acoustical Science and Technology, 40*(6), 406–409.

Child, R., Gray, S., Radford, A., & Sutskever, I. (2019). Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509

Choromanski, K. M., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J. Q., Mohiuddin, A., Kaiser, L., Belanger, D. B., Colwell, L. J., & Weller, A. (2021). Rethinking attention with performers. In *International conference on learning representations (ICLR).*

Chung, Y.-A., & Glass, J. (2018). Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech. In *Interspeech.*

Conneau, A., Baevski, A., Collobert, R., Mohamed, A., & Auli, M. (2021). Unsupervised cross-lingual representation learning for speech recognition. In *Interspeech.*

Dao, T., Fu, D., Ermon, S., Rudra, A., & Ré, C. (2022). Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Advances in neural information processing systems (NIPS).*

Dao, T., Fu, D.Y., Saab, K.K., Thomas, A.W., Rudra, A., & Ré, C. (2022). Hungry hungry hippos: Towards language modeling with state space models. arXiv preprint arXiv:2212.14052

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the conference of the North American Chapter of the Association for computational linguistics: Human language technologies.*

Dunbar, E., Bernard, M., Hamilakis, N., Nguyen, T. A., Seyssel, M., Rozé, P., Rivière, M., Kharitonov, E., & Dupoux, E. (2021). *T*he zero resource speech challenge 2021: Spoken language modelling. In *Interspeech.*

Ericsson, L., Gouk, H., Loy, C. C., & Hospedales, T. M. (2022). Self-supervised representation learning: Introduction, advances, and challenges. *IEEE Signal Processing Magazine, 39*(3), 42–62.

Evain, S., Nguyen, H., Le, H., Boito, M. Z., Mdhaffar, S., Alisamir, S., Tong, Z., Tomashenko, N., Dinarelli, M., Parcollet, T., Allauzen, A., Esteve, Y., Lecouteux, B., Portet, F., Rossato, S., Ringeval, F., Schwab, D., & Besacier, L. (2021). LeBenchmark: A reproducible framework for assessing self-supervised representation learning from speech. In *Interspeech.*

Gao, Y., Fernandez-Marques, J., Parcollet, T., Mehrotra, A., & Lane, N. D. (2022). Federated self-supervised speech representations: Are we there yet? arXiv preprint arXiv:2204.02804

Gaol, Y., Fernandez-Marques, J., Parcollet, T., Gusmao, P. P., & Lane, N. D. (2023). Match to win: Analysing sequences lengths for efficient self-supervised learning in speech and audio. In *IEEE spoken language technology workshop (SLT)*

Graves, A. (2012). Supervised sequence labelling with recurrent neural networks. *Studies in Computational Intelligence, 385*, 1–131.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., Kavukcuoglu, K., Munos, R., & Valko, M. (2020). Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in neural information processing systems (NIPS)*

Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., & Pang, R. (2020). Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech*.

Hsu, W.-N., Bolte, B., Tsai, Y.-H.H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29*, 3451–3460.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). Lora: Low-rank adaptation of large language models. In *International conference on learning representations (ICLR)*.

Huang, W., Zhang, Z., Yeung, Y. T., Jiang, X., & Liu, Q. (2022). Spiral: Self-supervised perturbation-invariant representation learning for speech pre-training. In *International conference on learning representations (ICLR)*.

Jang, E., Gu, S., & Poole, B. (2016). Categorical reparameterization with Gumbel–softmax. In *International conference on learning representations (ICLR)*.

Kahn, J., Riviere, M., Zheng, W., Kharitonov, E., Xu, Q., Mazaré, P.-E., Karadayi, J., Liptchinsky, V., Collobert, R., Fuegen, C., Likhomanenko, T., Synnaeve, G., Joulin, A., Mohamed, A., & Dupoux, E. (2020). Librilight: A benchmark for ASR with limited or no supervision. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Karimi Mahabadi, R., Henderson, J., & Ruder, S. (2021). Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in neural information processing systems (NIPS)*.

Kitaev, N., Kaiser, Ł., & Levskaya, A. (2020). Reformer: The efficient transformer. arXiv preprint arXiv:2001.04451

Lai, C.-I., Chuang, Y.-S., Lee, H.-Y., Li, S.-W., & Glass, J. (2021). Semi-supervised spoken language understanding via self-supervised speech and language model pretraining. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Lai, C.-I.J., Zhang, Y., Liu, A.H., Chang, S., Liao, Y.-L., Chuang, Y.-S., Qian, K., Khurana, S., Cox, D., & Glass, J. (2021). Parp: Prune, adjust and re-prune for self-supervised speech recognition. In *Advances in neural information processing systems (NIPS)*.

Le, D., Zhang, X., Zheng, W., Fügen, C., Zweig, G., & Seltzer, M.L. (2019). From senones to Chenones: Tied context-dependent graphemes for hybrid speech recognition. In *IEEE automatic speech recognition and understanding workshop (ASRU)*.

Lee, Y., & Jang, K., Goo, J., Jung, Y., & Kim, H.-R. (2022). Fithubert: Going thinner and deeper for knowledge distillation of speech self-supervised learning. In *Interspeech*.

Lee-Thorp, J., Ainslie, J., Eckstein, I., & Ontanon, S. (2022). Fnet: Mixing tokens with Fourier transforms. In *Proceedings of the conference of the North American Chapter of the association for computational linguistics: Human language technologies*.

Lester, B., Al-Rfou, R., & Constant, N. (2021). *The power of scale for parameter-efficient prompt tuning*. EMNLP.

Li, X.L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the annual meeting of the association for computational linguistics and the international joint conference on natural language processing*.

Lialin, V., Deshpande, V., & Rumshisky, A. (2023). Scaling down to scale up: A guide to parameter-efficient fine-tuning. arXiv preprint arXiv:2303.15647

Lin, T.-Q., Lee, H.-y., & Tang, H. (2022). Melhubert: A simplified Hubert on Mel spectrogram. arXiv preprint arXiv:2211.09944

Liu, A. H., Chang, H.-J., Auli, M., Hsu, W.-N., & Glass, J. R. (2023). Dinosr: Self-distillation and online clustering for self-supervised speech representation learning. In *Advances in neural information processing systems (NIPS)*.

Maekawa, A., Kobayashi, N., Funakoshi, K., & Okumura, M. (2023). Dataset distillation with attention labels for fine-tuning BERT. In *Proceedings of the 61st annual meeting of the association for computational linguistics*.

Mehta, H., Gupta, A., Cutkosky, A., & Neyshabur, B. (2022). Long range language modeling via gated state spaces. arXiv preprint arXiv:2206.13947

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaiev, O., Venkatesh, G., & Wu, H. (2018). Mixed precision training. In *International conference on learning representations (ICLR)*.

Mohamed, A., Lee, H.-Y., Borgholt, L., Havtorn, J. D., Edin, J., Igel, C., Kirchhoff, K., Li, S.-W., Livescu, K., Maaløe, L., Sainath, T. N., & Watanabe, S. (2022). Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing, 16*(6), 1179–210.

Moumen, A., & Parcollet, T. (2023). Stabilising and accelerating light gated recurrent units for automatic speech recognition. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Nguyen, T.A., Seyssel, M., Rozé, P., Rivière, M., Kharitonov, E., Baevski, A., Dunbar, E., & Dupoux, E. (2020). The zero resource speech benchmark 2021: Metrics and baselines for unsupervised spoken language modeling. In *NeuRIPS workshop on self-supervised learning for speech and audio processing*.

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Parcollet, T., Dalen, R., Zhang, S., & Bhattacharya, S. (2023). Sumformer: A linear-complexity alternative to self-attention for speech recognition. arXiv preprint arXiv:2307.07421

Parcollet, T., Zhang, S., Dalen, R., Ramos, A.G.C., & Bhattacharya, S. (2023). On the (in) efficiency of acoustic feature extractors for self-supervised speech representation learning. In *Interspeech*.

Park, D.S., Zhang, Y., Chiu, C.-C., Chen, Y., Li, B., Chan, W., Le, Q.V., & Wu, Y. (2020). Specaugment on large scale datasets. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., & Le, Q. V. (2019). Specaugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*.

Pasad, A., Chou, J.-C., & Livescu, K. (2021). Layer-wise analysis of a self-supervised speech representation model. In *IEEE automatic speech recognition and understanding workshop (ASRU)*.

Pasad, A., Shi, B., & Livescu, K. (2023). Comparative layer-wise analysis of self-supervised speech models. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Peng, Y., Kim, K., Wu, F., Sridhar, P., & Watanabe, S. (2023). Structured pruning of self-supervised pre-trained models for speech recognition and understanding. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Peng, Y., Sudo, Y., Muhammad, S., & Watanabe, S. (2023). Dphubert: Joint distillation and pruning of self-supervised speech models. In *Interspeech*.

Poli, M., Massaroli, S., Nguyen, E., Fu, D.Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., & Ré, C. (2023). Hyena hierarchy: Towards larger convolutional language models. arXiv preprint arXiv:2302.10866

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research, 21*(140), 1–67.

Ravanelli, M., Brakel, P., Omologo, M., & Bengio, Y. (2018). Light gated recurrent units for speech recognition. *IEEE Transactions on Emerging Topics in Computational Intelligence,2*

Reed, C.J., Yue, X., Nrusimha, A., Ebrahimi, S., Vijaykumar, V., Mao, R., Li, B., Zhang, S., Guillory, D., Metzger, S., Keutzer, K., & Darrell, T. (2022). Self-supervised pretraining improves self-supervised pretraining. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*.

Roy, A., Saffar, M., Vaswani, A., & Grangier, D. (2021). Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*.

Sadhu, S., He, D., Huang, C.-W., Mallidi, S.H., Wu, M., Rastrow, A., Stolcke, A., Droppo, J., & Maas, R. (2021). Wav2vec-C: A self-supervised model for speech representation learning. In *Interspeech.*

San, N., Bartelds, M., Browne, M., Clifford, L., Gibson, F., Mansfield, J., Nash, D., Simpson, J., Turpin, M., Vollmer, M., Wilmoth, S., & Jurafsky, D. (2021). Leveraging pre-trained representations to improve access to untranscribed speech from endangered languages. In *IEEE automatic speech recognition and understanding workshop (ASRU)*

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). Distilbert, a distilled version of Bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108

Schlag, I., Irie, K., & Schmidhuber, J. (2021). Linear transformers are secretly fast weight programmers. In *International conference on machine learning (ICML)*. PMLR.

Schneider, S., Baevski, A., Collobert, R., & Auli, M. (2019). wav2vec: Unsupervised pre-training for speech recognition. In *Interspeech.*

Seltzer, M.L., Yu, D., & Wang, Y. (2013). An investigation of deep neural networks for noise robust speech recognition. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Seo, S., Kwak, D., & Lee, B. (2022). Integration of pre-trained networks with continuous token interface for end-to-end spoken language understanding. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Shi, Y., Paige, B., Torr, P., & Siddharth, N. (2020). Relating by contrasting: A data-efficient framework for multimodal generative models. In *International conference on learning representations (ICLR)*.

Stafylakis, T., Mošner, L., Kakouros, S., Plchot, O., Burget, L., & Ćernockỳ, J. (2022). Extracting speaker and emotion information from self-supervised speech models via channel-wise correlations. In *IEEE Spoken language technology workshop (SLT)*.

Sung, Y.-L., Cho, J., & Bansal, M. (2022). LST: Ladder side-tuning for parameter and memory efficient transfer learning. In *Advances in neural information processing systems (NIPS).*

Tay, Y., Dehghani, M., Bahri, D., & Metzler, D. (2022). Efficient transformers: A survey. *ACM Computing Surveys,6*

Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., & Li, Y. (2022). Maxvit: Multi-axis vision transformer. In *European conference on computer vision*.

Tyagi, S., & Sharma, P. (2020). Taming resource heterogeneity in distributed ml training with dynamic batching. In *IEEE international conference on autonomic computing and self-organizing systems (ACSOS)*.

Vyas, A., Hsu, W.-N., Auli, M., & Baevski, A. (2022). On-demand compute reduction with stochastic wav2vec 2.0. arXiv preprint arXiv:2204.11934

Wang, R., Bai, Q., Ao, J., Zhou, L., Xiong, Z., Wei, Z., Zhang, Y., Ko, T., & Li, H. (2022). Lighthubert: Lightweight and configurable speech representation learning with once-for-all hidden-unit BERT. In *Interspeech*.

Wang, S., Li, B.Z., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-attention with linear complexity. arXiv preprint arXiv:2006.04768

Wang, Y., Li, J., Wang, H., Qian, Y., Wang, C., & Wu, Y. (2022). Wav2vec-switch: Contrastive learning from original-noisy speech pairs for robust speech recognition. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Wang, Y., Mohamed, A., Le, D., Liu, C., Xiao, A., Mahadeokar, J., Huang, H., Tjandra, A., Zhang, X., Zhang, F., Fuegen, C., Zweig, G., & Seltzer, M. (2020). Transformer-based acoustic modeling for hybrid speech recognition. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Wang, S., Nguyen, J., Li, K., & Wu, C.-J. (2023). Read: Recurrent adaptation of large transformers. arXiv preprint arXiv:2305.15348

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop blackbox NLP: Analyzing and interpreting neural networks for NLP.*

Wang, S., Zhou, L., Gan, Z., Chen, Y.-C., Fang, Y., Sun, S., Cheng, Y., & Liu, J. (2021). Cluster-former: Clustering-based sparse transformer for question answering. In *Proceedings of the annual meeting of the association for computational linguistics and the international joint conference on natural language processing*.

Wang, T., Zhu, J.-Y., Torralba, A., & Efros, A.A. (2018). Dataset distillation. arXiv preprint arXiv:1811.10959

Wu, F., Kim, K., Pan, J., Han, K.J., Weinberger, K.Q., & Artzi, Y. (2022). Performance-efficiency trade-offs in unsupervised pre-training for speech recognition. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Wu, Z., Liu, Z., Lin, J., Lin, Y., & Han, S. (2020). Lite transformer with long-short range attention. In *International conference on learning representations (ICLR)*.

Wu, C.-J., Raghavendra, R., Gupta, U., Acun, B., Ardalani, N., Maeng, K., Chang, G., Aga, F., Huang, J., Bai, C., Gschwind, M., Gupta, A., Ott, M., Melnikov, A., Candido, S., Brooks, D., Chauhan, G., Lee, B., Lee, H.-H., … Hazelwood, K. (2022). Sustainable AI: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems, 4*, 795–813.

Xie, Q., Luong, M.-T., Hovy, E., & Le, Q.V. (2020). Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Yang, B., Wang, L., Wong, D.F., Chao, L.S., & Tu, Z. (2019). Convolutional self-attention networks. In *Proceedings of the conference of the North American chapter of the association for computational linguistics: Human language technologies.*

Yang, S.-w., Chi, P.-H., Chuang, Y.-S., Lai, C.-I.J., Lakhotia, K., Lin, Y.Y., Liu, A.T., Shi, J., Chang, X., Lin, G.-T., Huang, T.-H., Tseng, W.-C., Lee, K.-T., Liu, D.-R., Huang, Z., Dong, S., Li, S.-W., Watanabe, S., Mohamed, A., & Lee, H.-Y. (2021). Superb: Speech processing universal performance benchmark. In *Interspeech*.

Yeh, S.-L., & Tang, H. (2022). Autoregressive co-training for learning discrete speech representations. In *Interspeech*.

Yu, A.W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M., & Le, Q.V. (2018). Qanet: Combining local convolution with global self-attention for reading comprehension. In *International conference on learning representations (ICLR)*.

Zaken, E.B., Goldberg, Y., & Ravfogel, S. (2022). Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th annual meeting of the association for computational linguistics*.

Zhai, S., Talbott, W., Srivastava, N., Huang, C., Goh, H., Zhang, R., & Susskind, J. (2021). An attention free transformer. arXiv preprint arXiv:2105.14103

Zhang, Q., Chen, M., Bukharin, A., He, P., Cheng, Y., Chen, W., & Zhao, T. (2023). Adaptive budget allocation for parameter-efficient fine-tuning. In *International conference on learning representations (ICLR)*.

Zhang, Y., Chen, G., Yu, D., Yao, K., Khudanpur, S., & Glass, J. (2016). Highway long short-term memory RNNS for distant speech recognition. In *IEEE international conference on acoustics, speech and signal processing (ICASSP)*.

Zhang, Y., Qin, J., Park, D.S., Han, W., Chiu, C.-C., Pang, R., Le, Q.V., & Wu, Y. (2020). Pushing the limits of semi-supervised learning for automatic speech recognition. arXiv preprint arXiv:2010.10504

Zhang, J.O., Sax, A., Zamir, A., Guibas, L., & Malik, J. (2020). Side-tuning: A baseline for network adaptation via additive side networks. In *European conference on computer vision*.