Big model only for hard audios: Sample dependent Whisper model selection for efficient inferences

Hugo Malard
LTCI, Télécom Paris
Institut Polytechnique de Paris

Salah Zaiem
LTCI, Télécom Paris
Institut Polytechnique de Paris

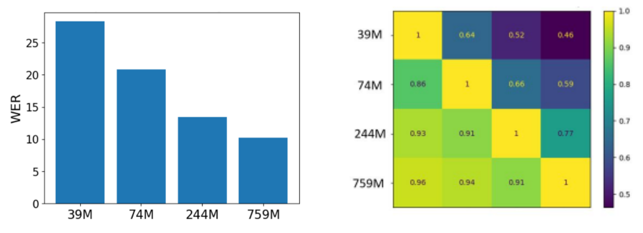
Robin Algayres
MBZUAI

***Abstract*—Recent progress in Automatic Speech Recognition (ASR) has been coupled with a substantial increase in model sizes, which may now hold billions of parameters, leading to slow inferences even with adapted hardware. In this context, several ASR models exist in various sizes, with different inference costs leading to different performance levels. Based on the observation that smaller models perform optimally on large parts of testing corpora, we propose to train a decision module, that would allow, given an audio sample, to use the smallest sufficient model leading to a good transcription. We apply our approach to two Whisper models with different sizes. By keeping the decision process computationally efficient, we build a decision module that allows substantial computational savings with reduced performance drops.**

I. Introduction

Recent progress in neural-based automatic speech recognition (ASR) has been driven by new modeling architectures, data collection, and processing but also by larger models that have recently exceeded a billion parameters [1], [2]. Such advances have promised enhanced accuracy and capabilities, yet they have also ushered in escalating computational demands. These ASR models are usually available in a certain range of sizes with varying performance levels. For instance, Whisper models [1] are available in 6 sizes from Tiny (39M parameters) to Large (1.5B parameters), Nvidia FastConformers [3] range from Large (118M parameters) to XXLarge (1.2B parameters) and self-supervised models like Hubert [4] or WavLM [5] are generally available in Base and Large versions. Systematically, following the deep learning trend across modalities, larger version models, even when trained on the same datasets, perform substantially better than their reduced-size counterparts. This is shown in Figure 1 (a), where the mean Word Error Rates (WER) of four Whisper models with different sizes on the test set of the CommonVoice dataset [6] are presented. The mean WER drops from 28.1 with the “Tiny” version to 10.2 with the “Medium”.

However, as shown in Figure 1 (b), this performance drop may not concern a significant part of the testing points. In this figure, every cell (i, j) shows the proportion of samples in the CommonVoice test set where model i performs better or equally to model j. For instance, the third cell in the first line (cell $(0, 2)$) states that for 52% of the testing samples, the Tiny model (39M) performs equally or better than the Small one (244M) while bearing more than 6 times fewer parameters. Based on this observation, this work explores whether we



(a) Absolute performances (b) Relative performances

Fig. 1. Absolute and relative performances of Whisper models on CommonVoice test set. The four models are Whisper Tiny (39M parameters), Base (74M), Small (244M) and Medium (759M). Each cell i, j in Figure (b) represents the percentage of utterances where the model i performs at least as well as the model j.

can predict if audio samples will fall into this category. By doing so, audio samples that would not benefit from the costly inference of a large model can be assigned to a smaller one in order to reduce the total computational load.

More precisely, this study aims to develop a *decision module* that, given an audio sample, chooses a Whisper model version that has the lowest inference cost without WER degradation. To keep experiments as simple as possible, and show effectiveness of the method, in this paper, we only focus on deciding if an audio sample should be decoded with Whisper Tiny (39M) or with Whisper Small (244M). These two model versions are relevant candidates as they exhibit high differences in both WER, inference costs, and latency.

A few works [7]–[9] have already attempted to choose among several ASR model versions using WER prediction. Given the textual output of an ASR model, they explored the prediction of the sentence-level WER. Yet, these methods are not aiming to reduce inference costs but rather to decide whether an audio sample should be reassigned to a more complex ASR model. Indeed, the most efficient techniques predict WER using full ASR pipelines based on either acoustic encoding and language-model-based beam search [8]. Such methods that rely on costly beam searches cannot be used in our case where the aim is to reduce the computational load.

Another close line of work is dynamic or early-exiting approaches. Instead of saving computation by choosing between separate ASR models, these methods have attempted to make forward passes lighter by skipping some of the last transformer layers of an ASR model [10]–[12]. The decision to exit is based on entropy or representation-similarity thresholds. However, early-exiting, as developed in these works, can only save layer computation in the ASR encoder, while, as shown in Table I, for attentional encoder-decoder architectures, most of the computations occur in the beam search decoding.

The closest work to our effort is from Lugosch *et al.* [13], who proposes to save computation cost at inference time by choosing between a large and a small ASR decoder. However, their method relies only on the log-likelihood of the encoder. We consider it as one of the baselines in this work.

This paper explores possibilities to build a decision module that allows efficient selection between different ASR model sizes while keeping high performance. Our contributions are threefold :

- We successfully reduce inference costs for a negligible WER degradation. In addition, our method can be used to interpolate between model sizes, saving the need for costly training of intermediate models.
- We explore different inputs and architectures for the decision module and compare them to several baselines and toplines.
- The codebase[1], developed within the SpeechBrain [14] framework, is released for further investigations.

II. Methods

This section describes our main pipeline represented in Figure 2. For a given audio sample, we first extract speech features and connect the output to a decider module. This latter is responsible for choosing to transcribe the audio with either a computationally cheap model, Whisper Tiny, or a more expensive one, Whisper Small. The computational cost of a model is measured using Multiply–ACcumulate (MAC) (the number of multiplications performed by the process).

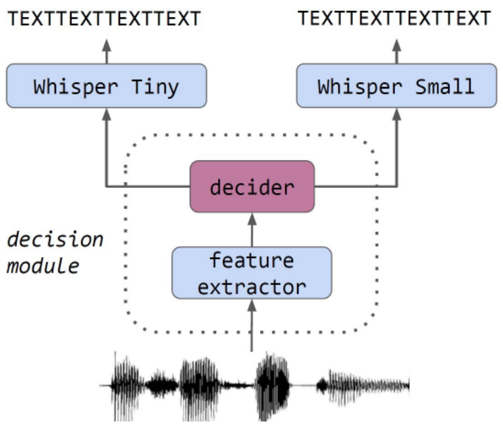


Fig. 2. The decision module is composed of a feature extractor that encodes the speech signal into latents that are given in input to the decider. Based on the output of the decider, the sample is transcribed either by Whisper Tiny or Whisper Small.

A. Feature Extractor

One of the questions raised in this work is which speech features are needed to predict how hard to transcribe is an audio sample. We explore two levels of representations, depending on their closeness to the raw signal waveform. We call low-level features hand-crafted spectral representations like Mel spectrograms while high-level features are typically the outputs of a transformer encoder trained with self-supervision [5], [15] or parallel text supervision [1]. Both type of features present their advantage for our double objective of minimizing transcription cost while keeping low WER: low-level features are cheap to compute and the high-level ones have proven to encode more disentangled phonetic features [16].

Model	Encoding	Beam-search decoding
Whisper Tiny	0.01T	0.22T
Whisper Small	0.13T	2.49T

TABLE I
Total MAC (multiply–accumulate) operations on CommonVoice test set for the encoder and the decoder of Whisper Tiny and Small using a beam search with n=8. T stands for Tera MACs (10e12))

We observed, however, that the computational cost of representing speech with high-level features is negligible compared to the cost of the full ASR pipeline. Indeed, in attention-based encoder-decoder models, like Whisper, the encoder maps speech to latent representations while the decoder converts them to text via beam search. While the encoding only costs one forward pass through the encoder, the autoregressive decoding costs multiple forward passes as the length of the output sequence increases. For instance, we present in Table I the computational cost of encoding and decoding the test set of CommonVoice [6] with either Whisper Tiny or Whisper Small. As expected, it appears that most of the computation is done in the beam-search decoding (22 and 19 times more MACs used in decoding than in encoding, for Whisper Tiny and Whisper Small respectively).

This suggests that decider performance, and not the computation cost of their extraction, should be the main criterion for input feature selection. Feature performance are evaluated in the following section.

B. The decider

As shown previously in Figure 1 (b), Whisper Tiny performs as well or even better than Whisper Small in 52% of the sentences in the CommonVoice dataset. The decider from Figure 1 is a neural model that aims to detect these elements to assign them to the smaller ASR processor. Specifically, for an audio sample a, the decider is trained on the output of the frozen feature extractor to predict 1 or 0 according to the following equation.

$$g_{\mathcal{M}_T,\mathcal{M}_S}(a) = \mathbb{1}_{WER(\mathcal{M}_T(a))>WER(\mathcal{M}_S(a))} \quad (1)$$

Where $\mathcal{M}_T$ and $\mathcal{M}_S$ are respectively Whisper Tiny and Whisper Small. It means that the model should learn to predict 0 if the WER obtained with $\mathcal{M}_T$ is lower than the one of $\mathcal{M}_S$.

In a computationally aware context, using a lightweight module for the decider is crucial, as its computational cost will be systematically paid at each inference. For instance, using a transformer stack for the decider would induce a high

[1]https://anonymous.4open.science/r/Big-model-only-for-hard-audios-2823

inference cost due to the high dimensionality of the output of the feature extractor. A convolutional stack, scaling linearly with the sequence length, does not suffer from those side effects. Moreover, the locality bias of the convolutions may be pertinent in this context since errors may occur at localized segments of the speech sample. Therefore, we opted for a one-dimensional ResNet [17] for the decider module architecture.

Instead of simply connecting the feature extractor output to the decider input, we learn a weighted sum of the feature extractor layers as in [18]. This method exploits the fact that different layers of a transformer stack encode different types of information from the audio signal [19].

III. Experiments

In this section, we describe the datasets and hyperparameters used for the training of the decider model. In addition, we present baselines and toplines that act as comparison points to our method.

A. Datasets and Settings

In this study, two datasets are considered: LibriSpeech [20] and CommonVoice 7.0 [6]. The LibriSpeech corpus is composed of read English speech recordings with 960 hours for training, two dev splits *dev-clean* and *dev-other* and two test splits *test-clean* and *test-other* of 5 hours each. CommonVoice is a collection of speech samples from worldwide users recording themselves with their own devices covering a large variety of age, gender and accents. The English part of the dataset presents roughly 1 260 hours of recorded audio.

The decider ResNet [17] is composed of 3 ResBlocks of two convolutional layers, each with 256 feature maps. The output of the ResNet is average-pooled before going through a linear layer with one sigmoid output neuron. The feature extractor remains frozen during the decoder training and a weight is learned for every layer. All the training details are available in the accompanying code repository.

B. Baselines and Toplines

We consider as baselines, three methods using threshold-based decisions. First, based on the well-known impact of noise on ASR quality, the output of a blind Signal-to-Noise Ratio (SNR) estimator [21] is used for the decision module. Here, the decider relies on a simple threshold, determined by Equal Error Rate (EER): it runs Whisper Small if the computed SNR is lower than the threshold and uses Whisper Tiny otherwise. Studies have highlighted the significant effect of English accents on ASR performance [22]. We therefore designed a second baseline that consists of using a pretrained accent detection model [23] to assign audios of rarer accents to the larger model. For this baseline, the decision module select Whisper Tiny if the English accent detected is either American, British, or Canadian.

Finally, inspired by Lugosch *and al.* [13], a third baseline explores the use of ASR encoder-decoder token probabilities as a confidence measure. Precisely, using Whisper Tiny, we perform full greedy decoding, compute the entropy of the output probabilities for each time step, and then aggregate with mean pooling over the time dimension. Here again, the decider is a threshold on entropy values that is set on a validation set as for the SNR baseline.

Feature extractor	**Decider**	**test-clean$\uparrow$**	**test-other$\uparrow$**	**CV test$\uparrow$**
SNR [21]	thresh.	50.7	47.2	47.0
Accent [23]	thresh.	n/a	n/a	52.0
$\mathcal{M}_T$ logit entropy	thresh.	64.4	63.7	64.5
Mel spectr.	ResNet	62.5	55.2	60.0
Wav2Vec2.0 Base	ResNet	52.3	57.7	63.9
$\mathcal{M}_T$ encoder	ResNet	65.1	65.0	66.4
$\mathcal{M}_S$ encoder	ResNet	**68.0**	**66.6**	**68.2**
WER $\mathcal{M}_T$	*thresh.*	*84.8*	*80.7*	*80.3*

TABLE II
F1-scores (higher the better) of the several decision modules on LibriSpeech and CommonVoice test sets. SNR, Accent and Logit Entropy are baseline models that only require to fit a threshold on a validation set. When the Decider is ResNet, a dedicated ResNet is trained on each of the different feature extractors. The last line is a topline model based on an oracle that provides the WER of Whisper Tiny. $\mathcal{M}_T$ and $\mathcal{M}_S$ are respectively Whisper Tiny and Small.

Regarding toplines, we propose two decision modules that are voluntarily unrealistic in order to show the potential computational savings in the ideal case. The first topline is an oracle that knows the best model to chose for any audio sample (i.e Whisper Small only if its WER for the given utterance is lower than the one of Whisper Tiny). For the second topline, we assume that the WER of Whisper Tiny is known in advance for any audio sample. The decider here is a threshold on WER values set on a validation set to determine when to use the larger Whisper model. The threshold is chosen such that it corresponds to the larger value for which there is no false negative (*i.e.* choosing "Tiny" while it performs less well than "Small").

IV. Results and Discussion

This section describes the performances of the different methods and their associated computational costs.

Decider	**Test-Clean$\uparrow$**	**Test-Other$\uparrow$**
ResNet	**68.0**	**66.6**
ResNet w/o weighted	66.4	66.0
Transformer	66.2	65.4
TL-Transformer	64.4	55.2

TABLE III
Ablation of the decider architecture: LibriSpeech results (F1-scores) for the ResNet with and without weighted inputs sum, transformer encoder, and time-wise layer-wise transformer

A. F1-score of decision modules

Table II presents the decision modules F1-scores. It consists, for a collection of audio samples, in the percentage of times a decision module correctly assigns audio samples to the appropriate Whisper model as defined in Equation 1.

The three first rows present the F1-score of the three baselines. The two first ones achieve close to random performances, which shows that neither SNR nor accent seem to capture the difficulty of the audio sample. For SNR, we hypothesize that this is due to the fact that the recordings come from relatively clean backgrounds, leading to very noisy and poorly informative estimation from the blind SNR estimator. On the contrary, logit-level entropy, although computationally costly gives significantly better results reaching an F1-score of 64.5. It seems to indicate that the internal states of the model contain useful information about the difficulty of decoding audio samples.

The second part of the table considers different feature inputs to the ResNet decider. As expected, higher-level features perform better than Mel spectrograms. The encoder of Whisper Small model performs better than Wav2vec2.0 in our setting with an F1-score of 68.2, compared to 63.9% for Wav2Vec2.0 features. This suggests that model-related features are the best suited to the decision task. Finally, the larger encoder performs better than the smaller one on the three considered test splits. Given this performance gap, and the low computation cost of encoding compared to decoding described in Section II-A, the remaining experiments will be conducted only using Whisper Small latent representations as input features to the decider.

The last line shows a topline. The oracle based on a threshold over the WER of Whisper Tiny produces strong results, showing that the WER of the smaller Whisper version can be used to allocate audio samples to large models efficiently. However, blind WER prediction remains a challenging and computationally costly endeavor [8].

We ablate the architectural choice of the decider model in Table III. First, removing the learned weighted sum of encoder layers slightly degrades the F1-score. Second, using a one-layer transformer network with a roughly equal number of parameters to the ResNet raises the decision computational cost, but also degrades F1-scores by a couple of percents. Finally, we implemented a small Time and Layer-wise (TL) transformer architecture which is composed of one transformer layer on the encoder layers outputs and another one on their pooled (time-wise) representations [24]. This approach performs worse than the simple one-layer transformer architecture.

B. WER/MACs Trade-off

Table IV shows the trade-off WER/MACs obtained with our pipeline compared to transcribing speech using Whisper models. It is important to note that the MACs column shows the cost of the full pipeline for transcribing the CommonVoice test set, including the cost of the decision module when there is one. Table IV starts with the WER/MACs of simply transcribing speech using the different Whisper models with beam search. Then, we include the logit entropy of Whisper Tiny, which is the best baseline from Table II. This baseline increases the WER by an absolute 0.7 points while reducing the model computational cost by $150G$. Our main pipeline comes next. It uses a decision module composed of Whisper Small encoder and Resnet. For the latter, we provide scores for two different decision thresholds: 0.3 and 0.5. We opt for the larger model when the sigmoid output is higher than the given threshold. By comparison with simply running Whisper Small, selecting a threshold of 0.5 on the sigmoid output of our decision module gives a 16% higher WER while resulting in a 35% decrease in MACs. Using a threshold of 0.3 increases the WER by an absolute 0.37 points while reducing the model computational cost by $310G$ MACs (*i.e.* 12% of the total cost). Finally, the last 2 lines show the hypothetical improvements that can be obtained using a perfect decider, or a threshold, based on a perfect estimation of the WER of $\mathcal{M}_T$. Not only do they allow for WER reduction, but they also reduce significantly the computational load. These toplines confirm the large potential of the approach and call for further research on model assignment.

Method	**WER↓**	**MACs↓**
$\mathcal{M}_T$	28.1	0.23T
$\mathcal{M}_B$	20.7	0.60T
$\mathcal{M}_S$	13.3	2.62T
$\mathcal{M}_T$ logit entropy	14.0	2.47T
encoder $\mathcal{M}_S$ + ResNet @0.5	15.4	1.72T
encoder $\mathcal{M}_S$ + ResNet @0.3	13.7	2.31T
WER $\mathcal{M}_T$ Oracle	*12.93*	*1.954T*
Oracle	*12.27*	*1.468T*

TABLE IV
AVERAGE WER AND MACs (LOWER THE BETTER) ASSOCIATED WITH EACH METHOD (INCLUDING THE MACs OF THE DECIDER) ON THE COMMONVOICE TEST SET. THE TABLE STARTS WITH THE PERFORMANCES OF $\mathcal{M}_T$, $\mathcal{M}_B$ AND $\mathcal{M}_S$ WHICH ARE FULL ASR PIPELINE OF RESPECTIVELY WHISPER TINY, BASE AND SMALL. $\mathcal{M}_T$ LOGIT ENTROPY IS OUR BASELINE. ENCODER $\mathcal{M}_S$+ RESNET IS OUR MAIN CONTRIBUTION FOR WHICH WE GIVE PERFORMANCES AT TWO DIFFERENT THRESHOLD VALUES (0.3 AND 0.5). THE LAST 2 LINES ARE OUR TOPLINES.

Figure 3 shows the performances (WER) and computational costs (MACs) of the intermediate models obtained using our main pipeline (i.e. Whisper Small encoder and ResNet) and varying the sigmod threshold. Almost all the points are under the plotted diagonal, which means that the resulting drop in performance (relative to the larger model) is systematically smaller than the gain in computational cost. Whisper Base is included in Figure 3 as it is an intermediate model between Whisper Tiny and Small. Its WER/MACs trade-off is only slightly better than our selection approach. This shows that the method presented yields nearly equivalent performance to that of an intermediate model trained entirely from scratch, saving costly retrainings.

C. Discussion

Failure of the baselines and the encoder layers being the best-performing input, tend to show that the errors are very dependent on the ASR model, rather than complexities inherent to the audio signal that would make any ASR model fail to transcribe.

To investigate this, we compute correlation values between the WER of a Conformer Large [25] model trained on LibriSpeech [2] and of a Wav2Vec2.0 Base model fine-tuned on LibriSpeech, with the WER of Whisper tiny, over the samples

[2]https://huggingface.co/speechbrain/asr-conformer-transformerlm-librispeech

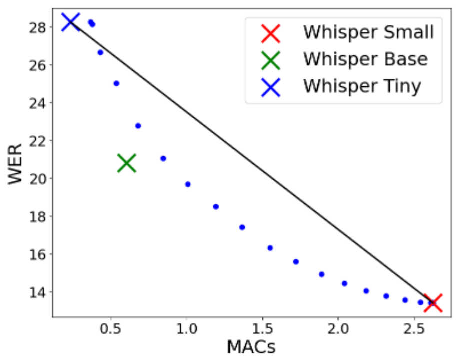


Fig. 3. WER/MACs values on the CommonVoice test set for different pipelines. The blue dots are the values for multiple thresholds using our best decision module (Whisper Small encoder and ResNet). The crosses correspond to the Whisper models Tiny, Base and Small respectively. Finally, the black line is a linear interpolation between the MACs and WER of Whisper Tiny and Whisper Small

of the LibriSpeech test sets (combined *clean* and *other*). The Pearson correlation coefficient between the WER of the Conformer model and the WER of Whisper Tiny reaches only 0.44, while the Spearman correlation is only 0.41. Similarly, these two correlation quantities between the WER of Wav2Vec2.0 and Whisper Tiny reach respectively 0.51 and 0.45. The low correlation values and the weak monotonic relationship seem to indicate that models have different intrinsic failure cases. It confirms, as our results have shown, that a successful model selection approach needs model-based inputs. This phenomenon may explains the recent success observed in the utilization of mixture of experts for ASR [26], [27].

V. Conclusion

In this study, we explored a new, computationally efficient approach, that selects for an audio sample the most efficient model among two of different sizes. It can be applied to interpolate between two trained models of fixed sizes without additional training, reducing the relative computational cost more than it degrades performances.

References

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," 2022.

[2] Y. Zhang, W. Han, J. Qin, Y. Wang, A. Bapna, Z. Chen, N. Chen, B. Li, V. Axelrod, G. Wang, Z. Meng, K. Hu, A. Rosenberg, R. Prabhavalkar, D. S. Park, P. Haghani, J. Riesa, G. Perng, H. Soltau, T. Strohman, B. Ramabhadran, T. Sainath, P. Moreno, C.-C. Chiu, J. Schalkwyk, F. Beaufays, and Y. Wu, "Google usm: Scaling automatic speech recognition beyond 100 languages," 2023.

[3] D. Rekesh, N. R. Koluguri, S. Kriman, S. Majumdar, V. Noroozi, H. Huang, O. Hrinchuk, K. Puvvada, A. Kumar, J. Balam, and B. Ginsburg, "Fast conformer with linearly scalable attention for efficient speech recognition," 2023.

[4] W. Hsu, B. Bolte, Y. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *CoRR*, vol. abs/2106.07447, 2021.

[5] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, and F. Wei, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *CoRR*, 2021.

[6] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," 2020.

[7] A. Ali and S. Renals, "Word error rate estimation without asr output: e-wer2," 2020.

[8] S. A. Chowdhury and A. Ali, "Multilingual word error rate estimation: e-wer3," 2023.

[9] A. K. Sheshadri, A. R. Vijjini, and S. Kharbanda, "Wer-bert: Automatic wer estimation with bert in a balanced ordinal classification paradigm," 2021.

[10] S. Zaiem, R. Algayres, T. Parcollet, S. Essid, and M. Ravanelli, "Fine-tuning strategies for faster inference using speech self-supervised models: A comparative study," 2023.

[11] J. W. Yoon, B. J. Woo, and N. S. Kim, "Hubert-ee: Early exiting hubert for efficient speech recognition," 2022.

[12] G. A. Wright, U. Cappellazzo, S. Zaiem, D. Raj, L. O. Yang, D. Falavigna, and A. Brutti, "Training dynamic models using early exits for automatic speech recognition on resource-constrained devices," *arXiv preprint arXiv:2309.09546*, 2023.

[13] L. Lugosch, D. Nowrouzezahrai, and B. H. Meyer, "Surprisal-triggered conditional computation with neural networks," 2020.

[14] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong *et al.*, "Speechbrain: A general-purpose speech toolkit," *arXiv preprint arXiv:2106.04624*, 2021.

[15] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[16] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," 2020.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[18] S. Yang, P. Chi, Y. Chuang, C. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G. Lin, T. Huang, W. Tseng, K. Lee, D. Liu, Z. Huang, S. Dong, S. Li, S. Watanabe, A. Mohamed, and H. Lee, "SUPERB: speech processing universal performance benchmark," *CoRR*, vol. abs/2105.01051, 2021.

[19] S. Zaiem, Y. Kemiche, T. Parcollet, S. Essid, and M. Ravanelli, "Speech Self-Supervised Representation Benchmarking: Are We Doing it Right?" in *Proc. INTERSPEECH 2023*, 2023.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[21] M. Lavechin, M. Métais, H. Titeux, A. Boissonnet, J. Copet, M. Rivière, E. Bergelson, A. Cristia, E. Dupoux, and H. Bredin, "Brouhaha: multitask training for voice activity detection, speech-to-noise ratio, and c50 room acoustics estimation," 2023.

[22] M. K. Ngueajio and G. Washington, "Hey ASR system! why aren't you more inclusive?" Springer Nature Switzerland, 2022.

[23] J. Zuluaga-Gomez, S. Ahmed, D. Visockas, and C. Subakan, "Commonaccent: Exploring large acoustic pretrained models for accent classification based on common voice," 2023.

[24] Y. Gong, S. Khurana, L. Karlinsky, and J. Glass, "Whisper-AT: Noise-Robust Automatic Speech Recognizers are Also Strong General Audio Event Taggers," in *Proc. INTERSPEECH 2023*, 2023.

[25] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," 2020.

[26] K. Hu, B. Li, T. N. Sainath, Y. Zhang, and F. Beaufays, "Mixture-of-expert conformer for streaming multilingual asr," 2023.

[27] Y. Kwon and S.-W. Chung, "Mole : Mixture of language experts for multi-lingual automatic speech recognition," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.