

Speech Self-Supervised Representation Benchmarking: Are We Doing it Right?

Salah Zaiem, Youcef Kemiche, Titouan Parcollet, Slim Essid,
Mirco Ravanelli
salah.zaiem@telecom-paris.fr

Interspeech 2023, Dublin, Ireland



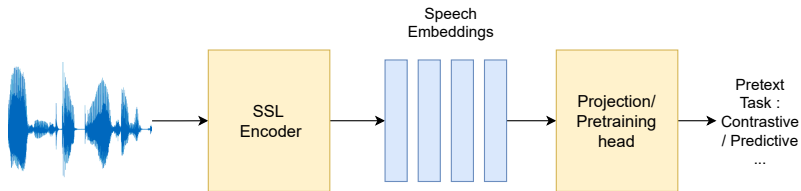
Outline

Introduction

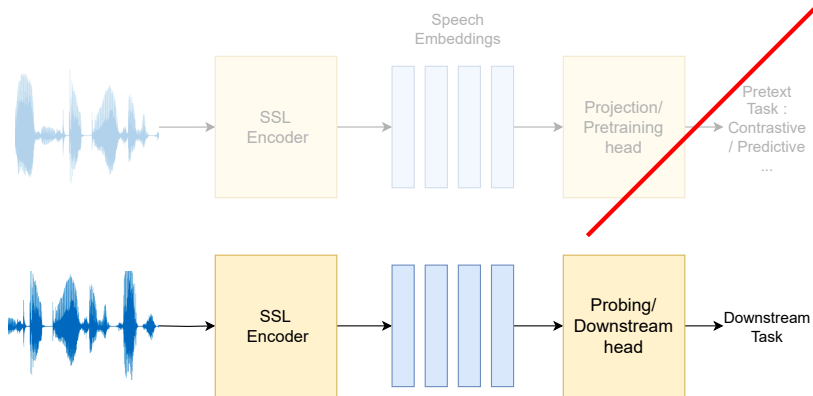
Experimental Setting

Results and Conclusions

Self supervised learning (SSL)



Self supervised learning (SSL)



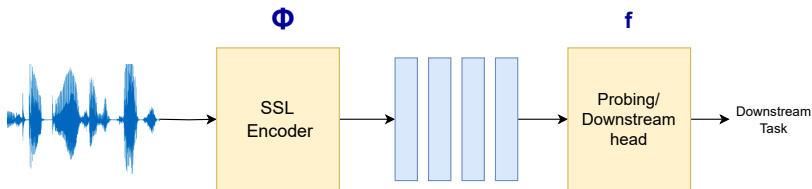
Benchmarking SSL Models

- ▶ Why? Plenty of SSL models, wide use of these representations in the recent literature, high cost of fine-tunings..
- ▶ How ? Evaluate the SSL representations on different speech downstream tasks using one fixed probing head.

Benchmarking SSL Models

Formally, a SSL pipeline consists of two models: a pre-trained encoder ϕ and a downstream probe f .

- ▶ ϕ is learned through solving a pretext task on large unlabeled speech datasets
- ▶ f is learned on the annotated downstream dataset.

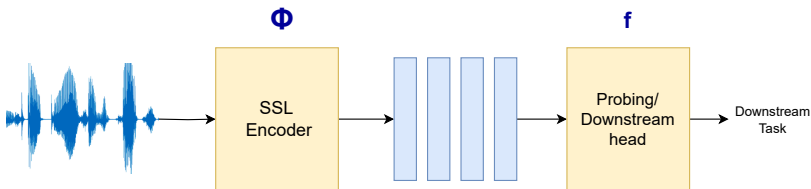


SUPERB Benchmark

The SUPERB (Yang and *al.*) benchmark has chosen for every considered downstream task T **one** probing family $\tilde{\mathcal{F}}_T$ (*i.e.* a downstream architecture) and shows for every considered SSL encoder ϕ a task error rate corresponding to:

$$\min_{f \in \tilde{\mathcal{F}}_T} E_t(f \circ \phi)$$

with $E_t(f \circ \phi)$ the test-set error rate of the full SSL pipeline.



SUPERB Benchmark

The SUPERB (Yang and *al.*) benchmark has chosen, in its "Constrained" track, for every considered downstream task T one probing family \mathfrak{F}_T (*i.e.* a downstream architecture) and shows for every considered SSL encoder ϕ a task error rate corresponding to:

$$\min_{f \in \mathfrak{F}_T} E_t(f \circ \phi)$$

Table 2: Evaluating various SSL representations on various downstream tasks. The numbers are collected with public-available checkpoints or codes, and we welcome researchers to re-submit the results to our online leaderboard.

	PR	KS	IC	SID	ER	ASR (WER)		QbE	SF		ASV	SD
	PER ↓	Acc ↑	Acc ↑	Acc ↑	Acc ↑	w/o ↓	w/LM ↓	MTWV ↑	F1 ↑	CER ↓	EER ↓	DER ↓
FBANK	82.01	8.63	9.10	8.5E-4	35.39	23.18	15.21	0.0058	69.64	52.94	9.56	10.05
PASE+ [16]	58.87	82.54	29.82	37.99	57.86	25.11	16.62	0.0072	62.14	60.17	11.61	8.68
APC [7]	41.98	91.01	74.69	60.42	59.33	21.28	14.74	0.0310	70.46	50.89	8.56	10.53
VQ-APC [32]	41.08	91.11	74.48	60.15	59.66	21.20	15.21	0.0251	68.53	52.91	8.72	10.45
NPC [33]	43.81	88.96	69.44	55.92	59.08	20.20	13.91	0.0246	72.79	48.44	9.4	9.34
Mockingjay [8]	70.19	83.67	34.33	32.29	50.28	22.82	15.48	6.6E-04	61.59	58.89	11.66	10.54
TERA [9]	49.17	89.48	58.42	57.57	56.27	18.17	12.16	0.0013	67.50	54.17	15.89	9.96
DeCoAR 2.0 [10]	14.93	94.48	90.80	74.42	62.47	13.02	9.07	0.0406	83.28	34.73	7.16	6.59
modified CPC [34]	42.54	91.88	64.09	39.63	60.96	20.18	13.53	0.0326	71.19	49.91	12.86	10.38
wav2vec [12]	31.58	95.59	84.92	56.56	59.79	15.86	11.00	0.0485	76.37	43.71	7.99	9.9
vq-wav2vec [13]	33.48	93.38	85.68	38.80	58.24	17.71	12.80	0.0410	77.68	41.54	10.38	9.93
wav2vec 2.0 Base [14]	5.74	96.23	92.35	75.18	63.43	6.43	4.79	0.0233	88.30	24.77	6.02	6.08
wav2vec 2.0 Large [14]	4.75	96.66	95.28	86.14	65.64	3.75	3.10	0.0489	87.11	27.31	5.65	5.62
HuBERT Base [35]	5.41	96.30	98.34	81.42	64.92	6.42	4.79	0.0736	88.53	25.20	5.11	5.88
HuBERT Large [35]	3.53	95.29	98.76	90.33	67.62	3.62	2.94	0.0353	89.81	21.76	5.98	5.75

SUPERB Benchmark

The SUPERB (Yang and *al.*) benchmark has chosen, in its "Constrained" track, for every considered downstream task T a probing family \mathfrak{F}_T (*i.e.* a downstream architecture) and shows for every considered SSL encoder ϕ a task error rate corresponding to:

$$\min_{f \in \mathfrak{F}_T} E_t(f \circ \phi)$$

* Params = parameter shared without fin

Method	Name	Description	URL	Params ↓	MACs ↓	(1) ↓	(2) ↓	(3) ↓	(4) ↓	Rank ↑	Score ↑
WavLM Large	Microsoft	M-P + VQ ...	🔗	3.166e+8	4.326e+12	3....	6....	1....	2....	25.8	1145
WavLM Base+	Microsoft	M-P + VQ ...	🔗	9.470e+7	1.670e+12	1....	2....	4....	8....	24.05	1106
WavLM Base	Microsoft	M-P + VQ ...	🔗	9.470e+7	1.670e+12	1....	2....	4....	8....	20.95	1019
data2vec Large	CI Tang	Masked G...	🔗	3.143e+8	4.306e+12	3....	6....	1....	2....	20.8	949
LightHuBERT Sta...	LightHuB...	Once-for-...	🔗	9.500e+7	-	-	-	-	-	20.1	959
HuBERT Large	paper	M-P + VQ	🔗	3.166e+8	4.324e+12	3....	6....	1....	2....	19.15	919
data2vec-aqc Base	Speech L...	Masked G...	🔗	9.384e+7	1.657e+12	1....	2....	4....	8....	19.05	935
CoBERT Base	ByteDanc...	Code Rep...	🔗	9.435e+7	1.660e+12	1....	2....	4....	8....	18	894
HuBERT Base	paper	M-P + VQ	🔗	9.470e+7	1.669e+12	1....	2....	4....	8....	17.75	941
wav2vec 2.0 Large	paper	M-C + VQ	🔗	3.174e+8	4.326e+12	3....	6....	1....	2....	17.7	914

Benchmarking SSL Models

However, this is only an approximation. Ideally, as proposed in the "unconstrained" track of SUPERB, the shown performance would be :

$$\min_{\mathfrak{F} \in \mathfrak{P}} \min_{f \in \mathfrak{F}} E_t(f \circ \phi)$$

with \mathfrak{P} the set of all probes families.

In the "Less Constrained" scenario, \mathfrak{P} is replaced with \mathfrak{C} the set of probes that respect a chosen capacity constraint.

The two tracks remain empty of submissions. Is the "Constrained" approximation good enough ?

Why is it important ?

- ▶ SSL representations have become very popular in the speech community for almost all tasks
- ▶ These benchmarks are used during the development of new SSL models. Improving the benchmarks improves the models

Question

How resilient are benchmarks to changes in the selected downstream probes ?

=> To provide an answer, we will test how varying downstream heads influences the ranking and relative performances of SSL representations.

Outline

Introduction

Experimental Setting

Results and Conclusions

Models

- ▶ 9 models (subset of SUPERB), picked mainly according to performance
- ▶ Acting directly on the waveform
- ▶ Different training losses (contrastive, pseudo-labels, teacher-student..)
- ▶ Base and large models (difference in size and training data)

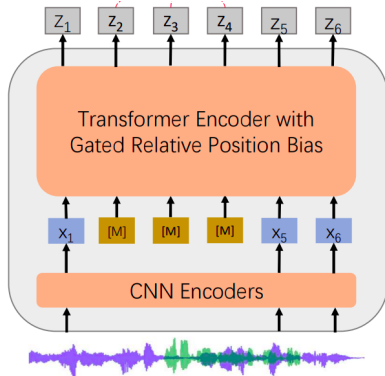


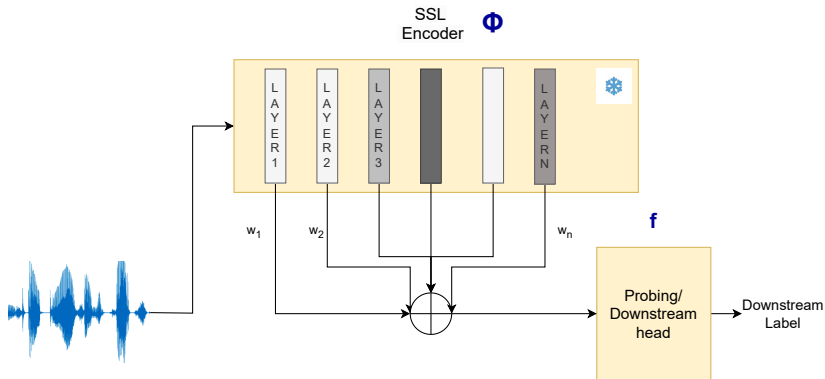
Fig 1 : Common architecture for the considered SSL Models

Benchmark : Tasks

- ▶ Automatic Speech Recognition : LibriSpeech *train-clean-100*, Buckeye, Basque, Welsh (Common Voice)..
- ▶ Intent Classification : SLURP, 18 scenarios
- ▶ Speaker Verification : VoxCeleb1
- ▶ Emotion Recognition : IEMOCAP, 4 classes

Benchmark : Global Setting

- ▶ Frozen SSL encoders, only the probing head are trained
- ▶ The needed information may be in various layers. The input representation is a weighted sum of the transformers layers of the SSL encoders. Weights are learnt, positive and sum to 1



Benchmark : Probing heads

First probing heads (SUPERB ones):

- ▶ Automatic Speech Recognition : 2 layers of BiLSTM, hidden size : 1024, CTC Loss
- ▶ Intent Classification, Emotion Recognition: Average time-pooling + linear classifier
- ▶ Speaker Verification : Xvector

Benchmark : Probing heads

Second probing heads :

- ▶ LibriSpeech : Encoder-Decoder Conformer
- ▶ CommonVoice low-resource languages : Two layered MLP
- ▶ Buckeye : ContextNet (Convolution-based)
- ▶ Emotion Recognition, Speaker Verification : ECAPA TDNN
- ▶ Intent Classification : BiLSTM encoder + linear classifier

Outline

Introduction

Experimental Setting

Results and Conclusions

First downstream results

Models /Tasks	SSL Params.	LibriSpeech train-100 ASR				Buckeye ASR		Welsh	Basque	ASV	ER	IC
Evaluation Metrics		WER ↓				WER ↓		WER ↓	WER ↓	EER ↓	Acc. ↑	Acc. ↑
First downstream architectures		LSTM				LSTM		LSTM	LSTM	Xvectors	Pool + Lin.	Pool + Lin.
		Clean	Other	Clean LM	Other LM	w/o LM	with LM	Welsh	Basque	ASV	ER	IC
DistilHuBERT	23.5M	13.99	34.91	9.96	28.26	35.59	28.29	53.20	46.78	9.1	65	46.6
Wav2vec 2.0 Base	95M	6.23	14.93	4.86	11.97	24.87	19.48	54.45	51.21	5.29	66.4	59.0
Wav2vec 2.0 Large	317.4M	3.72	9.25	3.13	7.48	20.72	16.11	45.42	37.98	5.69	69.3	66
HuBERT Base	94.7M	6.24	15.03	5.03	12.31	45.53	26.51	52.92	46.91	4.50	67.5	53.8
HuBERT Large	316.6M	3.57	8.12	2.90	6.59	51.30	33.10	51.21	46.15	5.20	71.3	69.9
WavLM Base+	94.7M	5.96	14.33	4.84	11.72	42.21	24.41	51.31	46.40	3.74	67.1	57.9
WavLM Large	316.6M	3.48	7.37	2.87	5.96	27.31	14.27	48.92	41.89	2.98	75.3	78.8
Data2vec Base	93.8M	5.30	13.79	4.03	10.97	37.26	30.50	54.00	46.37	5.43	63.0	56.9
Data2vec Large	314.3M	3.10	6.50	2.58	5.38	22.63	18.63	44.32	38.23	4.89	64.1	69.8
Probe size and inference metrics												
Downstream Parameters Base		39.9M				39.9M		40.3M	40.3M	7.0M	3.1k	13.8k
Downstream Parameters Large		42M				42M		42.4M	42.4M	7.7M	4.1k	18.4k

- ▶ Different ASR tasks lead to very **different** rankings
- ▶ Large versions are systematically better performing
- ▶ Except for ASR, the sizes of the probing heads are **small** compared to SSL encoders

Second downstream results

Models /Tasks	SSL Params.	LibriSpeech train-100 ASR				Buckeye ASR		Welsh	Basque	ASV	ER	IC
Evaluation Metrics		WER ↓				WER ↓		WER ↓	WER ↓	EER ↓	Acc. ↑	Acc. ↑
Second downstream architectures		Conformer				ContextNet		Lin.	Lin	ECAPA	ECAPA	LSTM + Lin.
		Clean	Other	Clean LM	Other LM	w/o LM	with LM	Welsh	Basque	ASV	ER	IC
DistilHuBERT	23.5M	14.97	36.51	11.54	31.41	58.56	43.61	80.78	77.04	2.85	72.4	74.9
Wav2vec 2.0 Base	95M	6.91	15.39	5.09	12.29	30.04	23.04	74.31	71.76	2.82	73.2	77.7
Wav2vec 2.0 Large	317.4M	4.32	9.25	3.58	7.03	23.92	18.68	75.45	78.48	3.17	68.4	79.0
HuBERT Base	94.7M	6.88	15.68	5.23	12.63	30.44	23.11	77.39	73.40	2.40	78.2	79.4
HuBERT Large	316.6M	3.96	8.60	3.10	6.88	39.39	31.57	71.58	60.24	3.84	71.5	80.1
WavLM Base+	94.7M	6.55	14.93	4.98	11.80	27.73	21.69	75.87	69.43	1.76	72.6	81.2
WavLM Large	316.6M	4.08	8.10	3.13	6.31	15.61	12.1	68.73	56.32	1.77	77.4	85.8
Data2vec Base	93.8M	5.85	14.32	4.53	12.52	40.53	33.45	77.49	75.26	3.75	72.0	73.4
Data2vec Large	314.3M	3.43	6.82	3.27	6.58	25.26	21.5	69.09	63.31	2.67	71.3	79.9
Probe size and inference metrics												
Downstream Parameters Base				11.2M		32.4M		1.9M	1.9M	9.2M	7.3M	42M
Downstream Parameters Large				11.2M		32.5M		2.3M	2.3M	9.8M	7.9M	44.1M

- ▶ Decoders with **larger** capacities
- ▶ Reduced difference in performance between Base and Large models

How different are the rankings and performances with the two sets of the probes ?

Correlations

Pearson and Spearman correlations are computed between the performances with the first and second sets of probing heads, for every task

Task	Pearson	Spearman	Mean DS1	Mean DS2	Diff (%)
LibriSpeech 1-2	0.99	0.97	5.8	6.48	-11.7
Buckeye	0.42	0.56	34.16	32.39	5.2
Welsh	0.59	0.62	50.64	74.52	-47.2
Basque	0.19	0.15	44.66	69.47	-55.6
VoxCeleb	0.47	0.75	5.2	2.78	46.5
Iemocap	0.22	0.34	67.66	73	7.9
Slurp	0.75	0.66	62.1	79.04	27.3

- ▶ Difference between “Mean DS1” and “Mean DS2” columns : performance highly sensitive to the chosen probing head. Improvements reach **46.5%** and **27.3%** respectively for ASV and IC
- ▶ **Low** correlation values, except for ASR on Librispeech

Main conclusions

First conclusion : Except for LibriSpeech ASR, current SSL benchmarking is not robust to the choice of the downstream probe

- ▶ Spearman correlation, between the two sets of performances, only reaches **0.34** and **0.66** for ER and IC respectively
- ▶ Other ASR tasks are heavily impacted by the probe change. Experiments with the the Buckeye corpus lead to 0.56 Spearman and 0.42 Pearson correlations => The high correlation is a LibriSpeech anomaly

Main conclusions

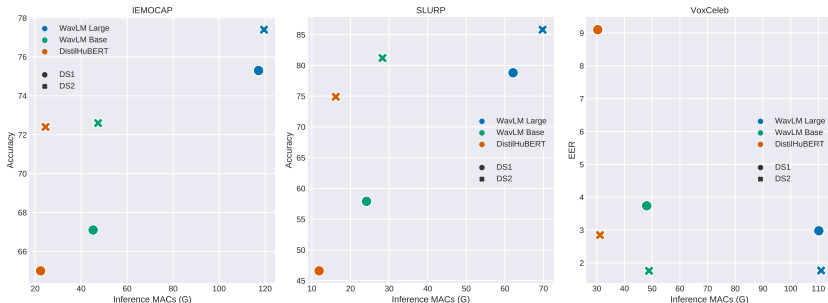
Second conclusion : Reduced difference in performance between large and base SSL models

- ▶ For IC, the mean absolute difference between the Base and Large versions performance drops from **14.23** to **3.28**
- ▶ For ER, while all four Large versions perform better than the Base ones when probed linearly, the new decoder **reverses** this order for all of them except WavLM

→ Probing with small decoders may be advantaging over-sized SSL encoders

Inference computations

Decoder computation are negligible compared to encoders ones. On the figures, the x-axis shift between circle and cross points show the computation cost of larger probing heads





Code sharing

The MP3S (for Multi-Probe Speech Self-Supervision) is now part of the SpeechBrain benchmarks sub-library.

SpeechBrain Benchmarks



 Tweet  SpeechBrain 196 members

★ Please, star our project on github (see top-right corner) if you appreciate our contribution to the community!

Welcome to the SpeechBrain Benchmarks repository! This repository is dedicated to housing a collection of benchmarks associated with the [SpeechBrain toolkit](#).

What are benchmarks? Benchmarks are standardized sets of recipes that enable users to measure the performance of specific models or techniques within a standardized environment. By utilizing these benchmarks, you can evaluate and compare the effectiveness of different approaches.

The SpeechBrain Benchmarks currently include the following:

- [CL_MASR](#) - A benchmark designed to assess continual learning techniques, specifically focusing on the continual learning of new languages for speech recognition.
- [MP3S](#) - A benchmark created to facilitate the fair assessment of self-supervised speech representations.



Take home messages

- ▶ Current self-supervised speech representations benchmarking is heavily biased by the choice of probing heads
- ▶ The limited capacity probes selected in the literature hinder the performance of small encoders, leading to over-sized SSL models

Thank you for your attention! Hope to see you on Wednesday, August, 23, for the “Analysis of Neural Speech Representations” session !