



# Speech Self-Supervised Representations Benchmarking: a Case for Larger Probing Heads

Salah Zaiem, Youcef Kemiche, Titouan Parcollet, Slim Essid, Mirco Ravanelli

## ► To cite this version:

Salah Zaiem, Youcef Kemiche, Titouan Parcollet, Slim Essid, Mirco Ravanelli. Speech Self-Supervised Representations Benchmarking: a Case for Larger Probing Heads. 2023. <hal-04216187>

**HAL Id: hal-04216187**

**<https://hal.science/hal-04216187v1>**

Preprint submitted on 23 Sep 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

# Speech Self-Supervised Representations Benchmarking: a Case for Larger Probing Heads

Salah Zaiem, LTCI, Télécom Paris, Institut Polytechnique de Paris, [salah.zaiem@telecom-paris.fr](mailto:salah.zaiem@telecom-paris.fr)  
 Youcef Kemiche, Capgemini, Hi! PARIS Engineering Team, Paris, France, [youcef.kemiche@capgemini.com](mailto:youcef.kemiche@capgemini.com)  
 Titouan Parcollet, Samsung AI Center Cambridge, United-Kingdom, [t.parcollet@samsung.com](mailto:t.parcollet@samsung.com)  
 Slim ESSID, LTCI, Télécom Paris, Institut Polytechnique de Paris, [slim.essid@telecom-paris.fr](mailto:slim.essid@telecom-paris.fr)  
 Mirco Ravanelli, Concordia University, Mila-Quebec AI Institute, Canada [mirco.ravanelli@concordia.ca](mailto:mirco.ravanelli@concordia.ca)

**Abstract**—Self-supervised learning (SSL) leverages large datasets of unlabeled speech to reach impressive performance with reduced amounts of annotated data. The high number of proposed approaches fostered the emergence of comprehensive benchmarks that evaluate their performance on a set of downstream tasks exploring various aspects of the speech signal. However, while the number of considered tasks has been growing, most proposals rely upon a single downstream architecture that maps the frozen SSL representations to the task labels. This study examines how benchmarking results are affected by changes in the probing head architecture. Interestingly, we found that altering the downstream architecture structure leads to significant fluctuations in the performance ranking of the evaluated models. Against common practices in speech SSL benchmarking, we evaluate larger-capacity probing heads, showing their impact on performance, inference costs, generalization and multi-level feature exploitation.

**Index Terms:** self-supervised learning, representation learning

## I. INTRODUCTION

Self-supervised learning (SSL) offers a compelling solution for benefiting from abundant unlabeled data to achieve notable performance improvements in various downstream tasks such as speech or speaker recognition. Numerous techniques have been introduced in the literature, such as predictive coding [1], [2], multi-task learning [3], [4], and contrastive learning approaches [5], [6]. Recently, self-supervised representations have emerged as indispensable tools for speech practitioners who face challenges due to insufficient annotations across an expanding range of tasks [7].

However, experimenting with large SSL models is a costly endeavor both in terms of time and computing. The proliferation of approaches for speech SSL [8] has, therefore, fomented the need for “universal” benchmarks evaluating their performance across multiple downstream tasks. These benchmarks should serve as a means to explore different facets of the speech signal, enabling practitioners to make informed decisions tailored to their specific use cases. Benchmarks also allow the research community to have a common field of comparison for the different proposed SSL techniques and identify areas for improvement. Consequently, there has been a growing proliferation of comprehensive benchmarks in recent years [9]–[11]. These benchmarks offer standardized frameworks for evaluating the effectiveness of speech SSL models and algorithms. They encompass a wide array of speech applications. Even within a single objective like automatic

speech recognition (ASR), they provide various linguistic, acoustic, and prosodic configurations [12].

In prevalent speech SSL benchmarks, the evaluation of self-supervised representations typically involves using downstream decoders that map the frozen representations to the final downstream labels. These downstream probes are generally chosen based on simplicity and limited capacities, such as linear probing for classification tasks or shallow vanilla recurrent neural networks for speech recognition [9]. However, we hypothesize that this benchmarking approach may harm the development of novel SSL technologies in two significant ways. Firstly, the popularity of the main benchmarks, such as SUPERB [9], has established the considered downstream probes as the standard evaluation setting for any new speech SSL model. The metrics used in these benchmarks also contribute to shape the development of new approaches. Consequently, there may be a tendency to discard models that perform poorly with the selected probes, even if they could potentially excel with other downstream architectures. Secondly, the simplicity of the probes contrasts with the increasing complexity of SSL encoders. Testing with low-capacity probes can lead to an unnecessary transfer of complexity from the probing head, which is intended to be task-specific, to the encoder, which is expected to be more general. This transfer can result in unnecessarily large self-supervised models, leading ultimately to compute-costly inferences [13]. For example, in computer vision, Dubois et al. [14] demonstrated that changing the probe family from linear to multi-layer perceptrons (MLP) leads to different optimal hyperparameter values of SSL models and enables smaller SSL representations.

One potential solution to address these limitations is to explore headless evaluation alternatives that are not tied to specific downstream probes. While a few intrinsic quality assessment metrics for speech embeddings have been proposed [15], their correlation with downstream performances is still uncertain [16]. In image classification, Garrido et al. [17] demonstrated a strong correlation between the rank of vision SSL representations and final downstream performance, though the latter performance is obtained using linear probes exclusively. Recognizing these challenges, SUPERB [9] offers two tracks where researchers can choose their own downstream probes, with or without capacity constraints on the probing architectures. Regrettably, these two tracks have yet to receive any submissions.

This paper builds on previously published findings [18] which diagnosed the dependence of benchmarks on the choice of probing heads. Given that our initial results showed that different probing heads lead to different rankings, we argue that it is important to re-question the current practice followed by prominent benchmarks, where a particular probe is fixed for each task, without a clear justification. In this sense, we extend our previous study with a thorougher assessment of the benefits of performing the benchmarks with more-capacitated probing heads. Precisely, four desired characteristics are assessed: full pipeline performance, inference efficiency, generalization ability and the exploitation of multi-level encoder features. On all these points, our study shows an advantage for higher-capacity probing heads. These ideas and results aim to reshape the way the SSL models are benchmarked, and indirectly, ultimately influence their design towards better rankings in these benchmarks. Hence, the contributions of this work are threefold:

- 1) We benchmark a set of published state-of-the-art SSL models on various speech tasks, varying the downstream decoders, showing that, except for ASR on Librispeech, the rankings and relative performance are highly impacted by a change in the set of downstream probes (Section II)
- 2) We provide an extensive study on the impact of selecting higher-capacity decoders on performance, generalization abilities, inference efficiency, and feature-level selection and exploitation. IV
- 3) We release the code base developed within the SpeechBrain library [19] for replication and to encourage further investigations and comparisons between models.<sup>1</sup> The clean and easy-to use code is released within the “Benchmarks” sub-library. We call it “MP3S” standing for “Multi-Probe Speech Self-Supervision”.

## II. BENCHMARKING SSL MODELS : DEFINITION AND PROTOCOL

This section formally describes the limitation faced by current speech SSL benchmarks and also details the experimental protocol devised to bring this issue to light.

### A. Problem definition

Formally, a SSL pipeline consists of two systems: a pre-trained encoder  $\phi$  and a downstream probe  $f$ .  $\phi$  is learned through solving a pretext task on unlabeled speech datasets (e.g., Libri-light [20] and LibriSpeech [21] have been popular choices in the literature), while  $f$  is learned for a considered downstream task with its corresponding annotated training dataset. In this framework, the SUPERB benchmark has chosen a probing family  $\mathfrak{F}_T$  (i.e. a downstream architecture with its hyperparameters, such as an MLP with given number of layers and hidden sizes) for every considered downstream task  $T$  and, for every considered SSL encoder  $\phi$ , it shows a task error rate corresponding to:

$$\min_{f \in \mathfrak{F}_T} E_t(f \circ \phi); \quad (1)$$

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

However, ideally, as proposed in the “*unconstrained*” track of SUPERB [9], the shown performance should be:

$$\min_{\mathfrak{F} \in \mathfrak{P}} \min_{f \in \mathfrak{F}} E_t(f \circ \phi); \quad (2)$$

with  $\mathfrak{P}$  the set of all probing families. More interestingly, in the “*constrained*” scenario, if we denote by  $\mathfrak{C}$  the set of probes that respect a chosen capacity constraint, then the performance of an encoder  $\phi$  could be expressed as follows:

$$\min_{\mathfrak{F} \in \mathfrak{P}} \min_{f \in \mathfrak{F} \cap \mathfrak{C}} E_t(f \circ \phi). \quad (3)$$

Unfortunately, this quantity cannot be computed, as it would require training a model with every known downstream architecture that respects capacity constraints, for each considered encoder and task.

In this study, we aim to investigate whether benchmarking based on the value obtained in Equation (1) provides a robust ranking that remains consistent across different probing families. To achieve this, we examine different probing families for each downstream task and analyze whether the rankings and relative differences obtained in the initial experiments remain consistent in the subsequent experiments.

### B. Self-supervised pretrained models

For our study, we focused on a subset of state-of-the-art models from the SUPERB benchmark due to their wide adoption within the community. We selected nine SSL models that extract representations directly from the waveform: Wav2vec 2.0 [1], HuBERT [2], WavLM<sup>2</sup> [22], and Data2Vec [23] in both their Base and Large versions. We also included DistilHuBERT [24], which is a distilled version of Hubert Base with four times fewer transformer layers. These models share the same frame rate, generating representations of dimension  $D$  every 20 ms of audio signal.  $D = 1,024$  for the “Large” versions and  $D = 768$  for “Base” ones and DistilHuBERT.

These models share similar Transformer-based architectures, but their pretraining pretext tasks vary. Wav2vec2.0 is trained using contrastive predictive coding (CPC), aiming to maximize mutual information between contextual features and predicted future samples. HuBERT and WavLM learn to map unlabeled audio to sequences of pseudo-labels generated through clustering previously generated representations. WavLM introduces training distortions to HuBERT enabling noise-invariant representations. Data2Vec, inspired by teacher-student approaches, employs a masked input view to predict latent representations of the unmasked input data, utilizing a self-distillation setup. We obtained all the pre-trained checkpoints from their respective HuggingFace (HF) official cards [25], except for Wav2vec2.0 Large, for which we used the Fairseq [26] checkpoint since the HF version underperformed compared to the results reported in SUPERB.

<sup>1</sup>github.com/speechbrain/benchmarks/tree/main/benchmarks/MP3S

<sup>2</sup>We used the Base+ version of WavLM, trained on 94k hours of speech data

### C. Downstream Tasks and Datasets

Speech SSL benchmarks attempt to assess universal speech representations by offering a diverse array of tasks that examine various facets of the speech signal. In line with this approach, we introduce seven tasks that cover phonetic, speaker-identity, emotional, and semantic dimensions.

**Speech Recognition Tasks.** Four speech recognition tasks are considered. For the first one, LibriSpeech [21] *train-clean-100/dev-clean* subsets are used for training and validation while *test-clean* and *test-other* are kept for testing. The Buckeye dataset [27] is considered as a second ASR task, allowing for testing the ability of the models with fewer labeled data and in a more complex spontaneous setting of English speech. The training, validation, and test splits used in our Buckeye experiments are available in the companion repository with the training set containing approximately 9.5 hours of audio and the test set 1.5 hour. For these two English ASR tasks, we present two sets of results based on the use or not of a language model (LM) during the decoding process. In the experiments labeled “Without LM,” we employ greedy decoding. Conversely, the “With LM” experiments utilize the official LibriSpeech 4-gram language model combined with shallow fusion to the acoustic model. Since low-resource languages are one of the main applications of SSL methods, two low-resource language tasks, extracted from the CommonVoice 11.0 [28] release, are considered: Welsh (*Cymraeg*) and Basque (*Euskera*). To ease reproducibility, we use the splits provided in the CommonVoice release: the Basque train set is 15.8-hour long, with 56 different speakers, while test and dev splits are 10.5 and 9.8-hour long. For Welsh, train, dev and test, splits are respectively, 11, 7.9 and 8 hour-long with 32 different speakers in the training set. The Word Error Rate (WER) serves as the error metric for all ASR tasks. In all these experiments, the probe is trained using the Connectionist Temporal Classification (CTC) loss at the character level.

**Automatic Speaker Verification (ASV).** The ASV task consists of a binary classification procedure aimed at determining whether speakers in a pair of utterances are the same. Similar to the SUPERB benchmark, we utilize the VoxCeleb1 train and test splits for this task [29]. It is worthwhile to note that the testing set may include speakers who were not present in the training set. The evaluation metric employed for ASV is the Equal Error Rate (EER).

**Emotion Recognition (ER).** For ER, we utilize the IEMOCAP dataset [30], which comprises 10,039 utterances from 10 distinct speakers. The objective of this task is to predict the emotional class of a speech utterance from four possible candidates: *neutral*, *happy*, *sad*, and *angry*. The reported performance represents the mean of 10 runs conducted through cross-validation on 10 folds, where each fold leaves out the data of one speaker for testing purposes.

**Intent Classification (IC).** While the SUPERB benchmark

evaluates the semantic content of SSL representations using the Speech Commands (SC) [31], we employ the more challenging SLURP dataset [32] for Intent Classification, as error rates with SC are extremely low. The SLURP collection consists of approximately 72,000 audio recordings that capture user interactions with a home assistant in single-turn scenarios. The IC task involves classifying each utterance into one of the 18 predefined scenarios, such as “calendar”, “email”, and “alarm”. Classification accuracy serves as the metric for both emotion recognition and intent classification tasks.

### D. Downstream Probes

This section offers a high-level description of the downstream probes employed in the study. For comprehensive replication of the experiments, detailed information regarding hyperparameters and architectural specifications can be found in the code repository.

**Global settings.** During the downstream training, the weights of the SSL encoder are kept frozen, learning solely the weights of the downstream decoder. Similarly to SUPERB, we observed that the last-layer representation may not always be optimal. Consequently, we, first, store the representations from all hidden layers of the pre-trained model. These hidden states are then weighted and summed to create the representation forwarded to the decoder. The weights are trained during the downstream process. In order to ensure the validity of our experimental setting, we first reproduced the downstream architectures used in SUPERB during the initial set of experiments. Then, we modified the probes by introducing simpler or more complex alternatives inspired by the relevant literature for each task.

**Speech recognition tasks.** In the initial set of experiments, aimed at replicating the SUPERB conditions, a vanilla 2-layer Bidirectional LSTM (BiLSTM) with 1,024 units is utilized. This BiLSTM is followed by a linear layer that maps the latent representations to characters. For the second set of downstream architectures, we employ an encoder-decoder Conformer architecture [33] for the LibriSpeech task. The downstream architecture consists of 12 encoder layers, 4 decoder layers, and 4 attention heads. For the Buckeye task, we employ the convolutional-based ContextNet architecture [34] with unit strides to maintain the frame rate of the SSL models. In the case of Welsh and Basque from CommonVoice, a two-layer dense neural network is employed to map each frame representation to the probabilities of the corresponding characters. Additionally, experiments using ContextNet with LibriSpeech are also conducted. The performance of ContextNet and Conformer architectures, which are close to the state-of-the-art on LibriSpeech, motivated their selection as downstream probes. Different probes are selected for ASR tasks to show that eventual variations in performance are not linked to a unique couple of probes.

**Automatic speaker verification.** In the first experiment, we use the X-vector architecture [35] with the AM-Softmax loss [38] for training speaker embeddings. Verification is

TABLE I

SSL BENCHMARKING RESULTS FOR ALL TASKS AND DOWNSTREAM ARCHITECTURES. THE NUMBER OF PARAMETERS OF THE SSL ENCODER AND THE PROBES IS SHOWN IN THE “PARAMS” ROWS AND COLUMNS. UPPER PART CORRESPONDS TO THE RESULTS OBTAINED USING THE FIRST SET OF PROBING HEADS WHILE THE BOTTOM PART SHOWS THESE OBTAINED WITH THE SECOND SET. PROBING HEADS ARE COMPILED IN TABLE II.

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	<b>20.72</b>	16.11	45.42	<b>37.98</b>	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	<b>14.27</b>	48.92	41.89	<b>2.98</b>	<b>75.3</b>	<b>78.8</b>
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	<b>3.10</b>	<b>6.50</b>	<b>2.58</b>	<b>5.38</b>	22.63	18.63	<b>44.32</b>	38.23	4.89	64.1	69.8
<b>Probe size and inference metrics</b>												
Downstream Parameters Base		39.9M				39.9M		40.3M	40.3M	7.0M	13.8k	3.1k
Downstream Parameters Large		42M				42M		42.4M	42.4M	7.7M	18.4k	4.1k
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	<b>78.2</b>	79.4
HuBERT Large	316.6M	3.96	8.60	<b>3.10</b>	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	<b>1.76</b>	72.6	81.2
WavLM Large	316.6M	4.08	8.10	3.13	<b>6.31</b>	<b>15.61</b>	<b>12.1</b>	<b>68.73</b>	<b>56.32</b>	1.77	77.4	<b>85.8</b>
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	<b>3.43</b>	<b>6.82</b>	3.27	6.58	25.26	21.5	69.09	63.31	2.67	71.3	79.9
<b>Probe size and inference metrics</b>												
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

TABLE II

PROBES SELECTED FOR THE DOWNSTREAM TRAININGS. MORE DETAILS CAN BE FOUND IN THE COMPANION REPOSITORY.

Task/Probing Head	First Set	Second Set
LibriSpeech ASR	BiLSTM	Conformer [33]
Buckeye ASR	BiLSTM	ContextNet [34]
CommonVoice Low-Resource ASR	BiLSTM	Linear
Automatic Speaker Verification	X-Vectors [35]	ECAPA-TDNN [36]
Emotion Recognition	Time-Pooling + Linear	ECAPA-TDNN [36]
Intent Classification	Time-Pooling + Linear	BiLSTM + Linear [37]

performed using cosine similarity backend. In the second experiment, we employ the ECAPA-TDNN neural network [36], which integrates time-delay neural networks and parallel attention mechanisms to capture temporal dependencies and achieve state-of-the-art results in speaker verification [36].

**Classification tasks.** Similar to SUPERB, in the initial set of experiments, we employ linear probing for the classification tasks, namely intent classification and emotion recognition. The representations are first averaged along the time axis and then passed through a linear classification layer. For the second downstream architecture, inspired by state-of-the-art approaches [39], we opt for ECAPA-TDNN for emotion recognition. As for intent classification, we follow published work [37] and utilize two layers of BiLSTM with a hidden size of 1,024, followed by a linear classifier. This approach allows for considering the order of frame representations, in contrast to using time-pooled features. While the cited

works ([36], [37], [39]) employ these architectures on-top of handcrafted features (generally log-mel spectrograms), we show in the following that they are still relevant when fed with self-supervised representations. Table II provides a summary of the probing heads selected for our experiments.

### III. BENCHMARKING RESULTS AND DISCUSSION

Table I presents the comprehensive benchmarking results for the different SSL models. The upper and lower sections of the table display the performance achieved by the first and second sets of downstream architectures, respectively. Additionally, the number of neural parameters is reported for both the SSL encoder and downstream decoders. For the latter, only two values are provided per task (*i.e.*, “Base” or “Large”) as this number only depends on the dimension of the encoder output representations ( $D = 1024$  for “Large” and  $D = 768$  for “Base”). In the initial set of experiments, we replicated the SUPERB benchmark conditions for two tasks: LibriSpeech and VoxCeleb1. Notably, our results exhibited a Pearson correlation of 0.99 and 0.97, respectively, with the corresponding results on the SUPERB leaderboard. This high correlation validates our successful replication of the benchmark settings.

To study the impact of a decoder change on the final performances, we compute, for every task, the Pearson and Spearman correlations between the performance metrics obtained with the first downstream architectures and those obtained with the second ones, and collect them in Table III. The Pearson

TABLE III

CORRELATIONS (PEARSON AND SPEARMAN) BETWEEN THE PERFORMANCES ACHIEVED WITH THE FIRST AND SECOND DOWNSTREAM PROBES ARE GIVEN FOR EACH TASK. THE NUMBER IN THE COLUMN NAME INDICATES WHETHER THE RESULTS CORRESPOND TO THE FIRST OR SECOND SET OF PROBING HEADS, AND “DS” STANDS FOR “DOWNSTREAM”. “MEAN” COLUMNS SHOW THE MEAN PERFORMANCE ACROSS ALL THE CONSIDERED SSL ENCODERS. THE “DIFF” COLUMN PRESENTS THE RELATIVE DIFFERENCE IN MEAN PERFORMANCE BETWEEN THE TWO ARCHITECTURES. THE “FBANKS” COLUMNS SHOW THE PERFORMANCE ON EVERY TASK WITH MEL SPECTROGRAMS AS INPUT REPRESENTATIONS. THE DIFFERENCE BETWEEN “MEAN DS” AND “FBANKS DS” OUTLINES THE PERFORMANCE GAIN IN % FROM USING SSL REPRESENTATIONS INSTEAD OF HANDCRAFTED ONES.

Task	Pearson	Spearman	Mean DS1	Mean DS2	Diff (%)	FBANKS DS1	FBANKS DS2
LibriSpeech 1-2	0.99	0.97	5.8	6.48	-11.7	22.56	8.91
Librispeech 1-3	0.99	0.98	5.8	7.03	-21.2	22.56	43.12
Buckeye ASR	0.42	0.56	34.16	32.39	5.2	54.17	78.90
Welsh	0.59	0.62	50.64	74.52	-47.2	99.62	> 100
Basque	0.19	0.15	44.66	69.47	-55.6	> 100	> 100
ASV	0.47	0.75	5.2	2.78	46.5	9.28	3.41
ER	0.22	0.34	67.66	73	7.9	48.51	65.7
IC	0.75	0.66	62.1	79.04	27.3	12.6	42.3

correlation evaluates the linear relationship between the two sets of metrics, while the Spearman one assesses the strength and direction of their monotonic relationship. Correlation metrics close to 1 imply proportional performances and similar rankings between the SSL models used with different probes, making the benchmark robust to the considered downstream change. Correlation metrics close to zero indicate no correlation between the results of the two sets of experiments.

All the models tested demonstrate competitive performances on every downstream task and with every related decoding architecture. With the notable exception of LibriSpeech, all the downstream tasks error metrics vary substantially with changing probes. The mean performance of the SSL candidates with the first and second downstream decoders is presented in the last three columns of Table III. Notably, we observe a significant sensitivity to the choice of decoder as replacing the SUPERB decoder results in relative improvements of up to 46.5% for ASV and 27.3% for IC. This demonstrates the substantial impact that the decoder selection has on the performance of the SSL models. Furthermore, the Spearman and Pearson correlation values computed between the performances with the first and second set of downstream probes are low, despite being positive. This suggests significant variations in relative performances and rankings when comparing the results obtained with the two different downstream decoders. For instance, the Spearman correlation coefficients for ER and IC are only 0.34 and 0.66, respectively. It is noteworthy that while the assessment of LibriSpeech performance appears to be robust to decoder changes, this does not hold true for other ASR tasks. In the case of the spontaneous English Buckeye corpus, there is a Spearman correlation of 0.56 and a Pearson correlation of 0.42, while the Basque task exhibits correlations, Pearson and Spearman, of only 0.19 and 0.15. The Buckeye ASR scenario is particular as changing the decoder from BiLSTM to ContextNet leads to improved results for some models and detrimental effects for others. Specifically, the best-performing model, WavLM Large using the second decoder, ranks only fourth when evaluated with the SUPERB settings.

However, we noticed a contrasting pattern in the rankings and performance of the considered SSL encoders on the ASR task using LibriSpeech *train-clean-100*, as shown in Table III. Unlike the other downstream tasks, the rankings and perfor-

TABLE IV  
WORD ERROR RATE (WER %) RESULTS OF LIBRISPEECH EXPERIMENTS ON THE TWO CONSIDERED TEST SPLITS WITH CONTEXTNET AS A THIRD DOWNSTREAM PROBE. “DS” STANDS FOR DOWNSTREAM.

Tasks \ Models	SSL Params	Clean	Other	Clean LM	Other LM
DistilHuBERT	23.5M	20.52	43.27	10.44	29.17
Wav2vec 2.0 Base	95M	7.24	15.66	4.73	11.21
Wav2vec 2.0 Large	317.4M	4.35	8.68	03.03	6.86
HuBERT Base	94.7M	7.31	16.00	4.60	11.11
HuBERT Large	316.6M	4.04	8.63	2.98	6.45
WavLM Base+	94.7M	6.73	15.33	4.52	10.84
WavLM Large	316.6M	4.09	8.43	2.94	6.15
Data2vec Base	93.8M	5.46	13.34	3.76	10.04
Data2vec Large	314.3M	<b>3.50</b>	<b>6.94</b>	<b>2.56</b>	<b>5.36</b>
<b>Probe size and inference metrics</b>					
Downstream Parameters Base				32.4M	
Downstream Parameters Large				32.5M	

mance only exhibit minor variations when the downstream decoder is changed. To validate this observation, we conducted additional experiments using a third downstream decoder, ContextNet, specifically for this task. The results of this supplementary experiment are presented in Table IV, and the correlation values between performances with the first probe and the ContextNet are shown in the second row of Table III. Similarly, no significant differences were observed in the ranking of the SSL candidates. For instance, in all three setups without LM decoding, DistilHuBERT consistently exhibits the lowest performance among the candidates. Furthermore, “Large” versions of the considered candidates consistently outperform their “Base” counterparts on this task, independently of the used probing head. Table III further confirms these findings, revealing high Spearman and Pearson correlations exceeding 0.97 for LibriSpeech, while the highest correlation value observed for other tasks is only 0.75. This discrepancy indicates that the SSL encoders might be biased towards the LibriSpeech ASR task, which is not unexpected given its prominent role as a benchmark dataset and its consistent inclusion in the pretraining process datasets. These results lead us to the conclusion that current SSL benchmarking is highly dependent on the choice of the downstream probes, with the notable exception of LibriSpeech ASR.

#### IV. ON LIMITED-CAPACITY PROBING HEADS

The first section has shown that the rankings and relative performances of the benchmarked self-supervised sys-

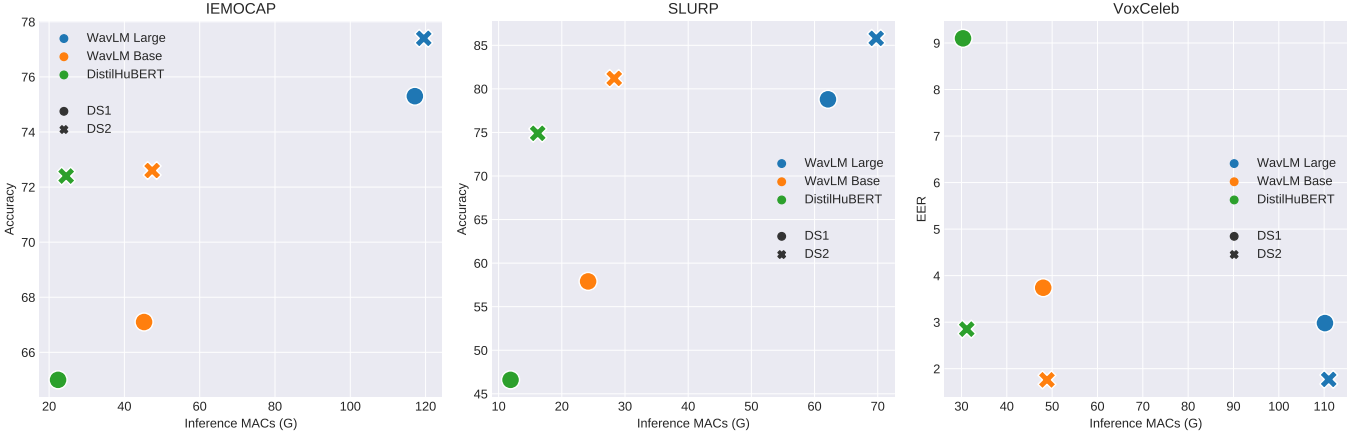


Fig. 1. Performance vs mean total inference cost metrics (in G-MACs) depending on the probing heads used for three models and three different downstream tasks. On all tasks, second downstream probes, larger in capacity, allow smaller SSL models to bridge the gap with bigger ones in term of accuracy with limited additional inference costs.  $DS(i)$  for  $i \in 1, 2$  corresponds to the results obtained with the  $i$ -th set of downstream probes.

tems are heavily impacted by a change in the downstream probing heads. The question that naturally arises is whether the common choice of probing heads is justified enough to discourage evaluating with other alternatives. The proposed downstream probes in the prominent SUPERB benchmark were selected based mainly on a simplicity criterion. Choosing simple probing heads is generally justified by the fact that it allows for evaluating only the quality of the pre-trained representations and not the downstream probes learning abilities. In this section, we will show that choosing limited-capacity decoders is not optimal. To prove it, and based on the previous experiments and further ones, we will show that larger probing heads: 1) lead to better performance; 2) reduce the error rate gaps between large and smaller SSL encoders, potentially leading to lower inference times; 3) enable the exploitation of multi-level features within the encoders; and 4) do not harm the generalization abilities of the full pipeline.

#### A. Performance and Inference Costs

This subsection elaborates two conclusions from the presented results and further computations of inference metrics. First, on most tasks, larger capacity decoders improve significantly the performance, allowing an optimal use of the pretrained representations. Second, larger-capacity probes enable smaller SSL encoders to bridge the performance gap with larger ones, eventually leading to faster inferences.

Concerning performance, Table III shows that except for the Buckeye ASR task, the mean performance is better with the probes with larger capacities, mainly for Speaker Verification and Intent Classification with respectively 46.5% and 27.3% relative performance improvements (for ASR tasks, the first probe, two layers of BiLSTM, is the largest probe in terms of number of parameters as shown in Table I). Decoders with more capacity seem naturally able to better exploit the benchmarked representations. For instance, time-pooling the frame-level representations before emotion or intent classification prevents the model from learning to use local or time-ordered signal clues, while it is possible with ECAPA-TDNN or a

layer of BiLSTM in the probing head. To know whether the performance increase is imputable to the representations or the probes, we compute the performance of the downstream probes using Mel-scaled spectrograms as the input representation. The spectrograms’ extraction is done similarly to the one provided as baseline in the SUPERB benchmark [9]. The results are shown in the last two columns of Table III. We can see, first, that the mean performance is significantly better using learned representations than hand-crafted Mel spectrograms, especially for ASR where the final WER is over 100 in three cases. For intent classification, the accuracy using SSL representations, is in average 5x better with the first probe and twice higher with the second probe. Moreover, apart for VoxCeleb, where two models perform worse than spectrograms with the second probe, all the representations benchmarked lead to better performances with all probes on all considered tasks. This shows that the lower error rates reached using larger decoders still depend on the quality of the input representations and that the levels of performance reached allow for an informed ranking of those.

Additionally, the findings presented in Table I shed light on an unexpected outcome when employing low-capacity decoders. With the first set of downstream architectures, the “Large” versions of SSL models consistently outperform their “Base” counterparts. However, this pattern does not hold true with higher-capacity decoders in the second set of probes. For example, the best performances in ASV and ER are achieved using WavLM Base+ and HuBERT Base, respectively. In the context of intent classification, changing the downstream decoder from linear to BiLSTMs results in a significant reduction in the mean absolute difference between the “Base” and “Large” versions’ performance, decreasing from 14.23 to 3.28. Again, for emotion recognition, although all four “Large” versions outperform their “Base” counterparts with linear probing, increasing the capacity of the probing head reverses this order for all models except WavLM. Additionally, in the case of ASV, DistilHuBERT achieves better results with an ECAPA decoder than the best-performing model (WavLM

Large) with an x-vector-based head, despite having more than 13 times fewer parameters. These findings suggest that using excessively small-capacity heads advantage larger SSL encoders and may have been leading to inflated model sizes.

Since the number of parameters does not present a full picture of the computations involved, the THOP library<sup>3</sup> is used to compute the number of Multiply–Accumulate operations (MACs) implied by the learned models. We compute exactly the mean number of MACs involved in inference (self-supervised feature extraction and downstream decoding) for every sample in the test set. Figure 1 shows the number of inference MACs for three models of different sizes and three considered downstream tasks: emotion recognition, intent classification, and speaker verification. For a fair comparison, we select the large models that perform the best on the considered task with the first downstream probe, along with its “Base” counterpart and DistilHuBERT as an even smaller competitor. First, on all three tasks, and for every model, the reached performance is systematically better with bigger decoders. Furthermore, the smallest encoder “DistilHuBERT”, while bearing 13 times less parameters than “Large” encoders, reaches a performance with the second decoder that is comparable to the best “Large” model with the first smaller downstream probe. Visually, for every considered model, the x-axis translation between the “DS1” (circle-shaped) and “DS2” points (cross-shaped) shows the MACs quantity increase induced by a bigger decoder head. While the BiLSTM-based decoder is visible on SLURP, the ECAPA-TDNN-based one seems negligible in the two other tasks compared to the self-supervision-based feature extraction costs. The three figures depict clearly both the high performance impact of a small boost in the decoder capacity and its low impact on the total computations needed for inference because of the large cost of feature extraction.

### B. Multi-level feature exploitation

The layer-wise content of speech self-supervised representations has been extensively probed throughout the literature [40], [41]. These studies generally assess the content with linear probes or with Canonical Correlation Analysis [42]. This subsection studies the impact of changing the probing head on the learned weighting of the layers of the models. It concludes that larger probing heads lead to a better exploitation of multi-level features in the considered self-supervised encoders.

As stated in section II-D, during fine-tuning, and in order to cover all the considered downstream tasks, a weighting of the SSL models’ layers is learned jointly to the probing heads parameters. With  $N$  the number of layers, 1 for the output of the convolutional front-end and  $N - 1$  transformer layers in the SSL encoders (3 in total for DistilHuBERT, 13 for “Base” models and 25 for “Large” ones),  $(P_i)_{i \in \{1, \dots, N\}}$  is a learned vector and  $W = \text{Softmax}(P)$  is the layer weighting vector. Let  $(R_i)_{i \in \{1, \dots, N\}}$  represent, for a given SSL encoder, the  $N$  matrices of intermediate embeddings of shape  $[T, D]$  with  $T$  the number of time frames (50 per second), and  $D$  the

dimension of the encoder learned representations. Then the input representation decoded by the probing head is:

$$R_{\text{input}} = \sum_{i=1}^N W_i R_i. \quad (4)$$

Figure 2 depicts the values (learned during every downstream training) of these weights for the four “Base” models considered in this work. The top part shows the learned weights with the first downstream probing heads, and the bottom part shows the second ones. First, it is very interesting to observe that the values of the learned weights seem to depend heavily on the SSL encoder pretraining task. While Data2Vec and Wav2Vec2.0 based, respectively, on masked language modeling and contrastive learning of quantized representations, display different weighting, HuBERT and WavLM, that have similar pretraining tasks, have very similar learned weighting for all the considered tasks, and with the two sets of downstream probes.

Second, it is important to note that the values of the learned weights are heavily impacted by changes in the considered probing head. This is especially the case for non-ASR tasks, and specifically for emotion recognition and intent classification. For these two tasks, with all the self-supervised encoders, only layers above the 9-th are selected with the linear probing approach. However, larger-capacity probes seem to be able to exploit low-level features.

For IEMOCAP, when using the first probing head, *i.e.* time-pooling followed by a linear classifier, the model relies on features from only one high-level layer (the last one for instance, for HuBERT and WavLM). On the contrary, probing with the ECAPA-TDNN—the second probing head considered here—spreads the weights across the different layers. In some cases, the last layers are barely weighted: Data2Vec, for instance, mainly uses the two first ones as shown in the first plot of the third row in Figure 2. This tends to indicate that the emotion recognition systems built using the linear probe may be exploiting linguistic content, while the second probe exploits mainly low-level emotion-related features. Concerning intent classification with the SLURP dataset, for HuBERT and WavLM, the main weight moves from the last layer to around the ninth one, while for Data2Vec, the LSTM-based decoder starts using multi-level features, including the first layer, *i.e.* the output of the convolutional front-end. We cannot easily draw a similar conclusion for ASR, where the high-level features are generally the closest to the phonetical content [40] and thus to the nature of the ASR task and seem to be naturally preferred by both the considered decoders. Finally, the VoxCeleb speaker recognition is always selecting low-level features, this is coherent with the layer-wise content probing literature [41], showing the loss of speaker information in high-level features of ASR-oriented self-supervised models.

Building on these observations, we argue that larger-capacity decoders enable the exploitation of multi-level features. In the case of intent classification and emotion recognition, this seems natural given that the first probes, time-pooling followed by a linear classifier, could only exploit features allowing for linearly separable downstream classes.

<sup>3</sup>github.com/Lyken17/pytorch-OpCounter



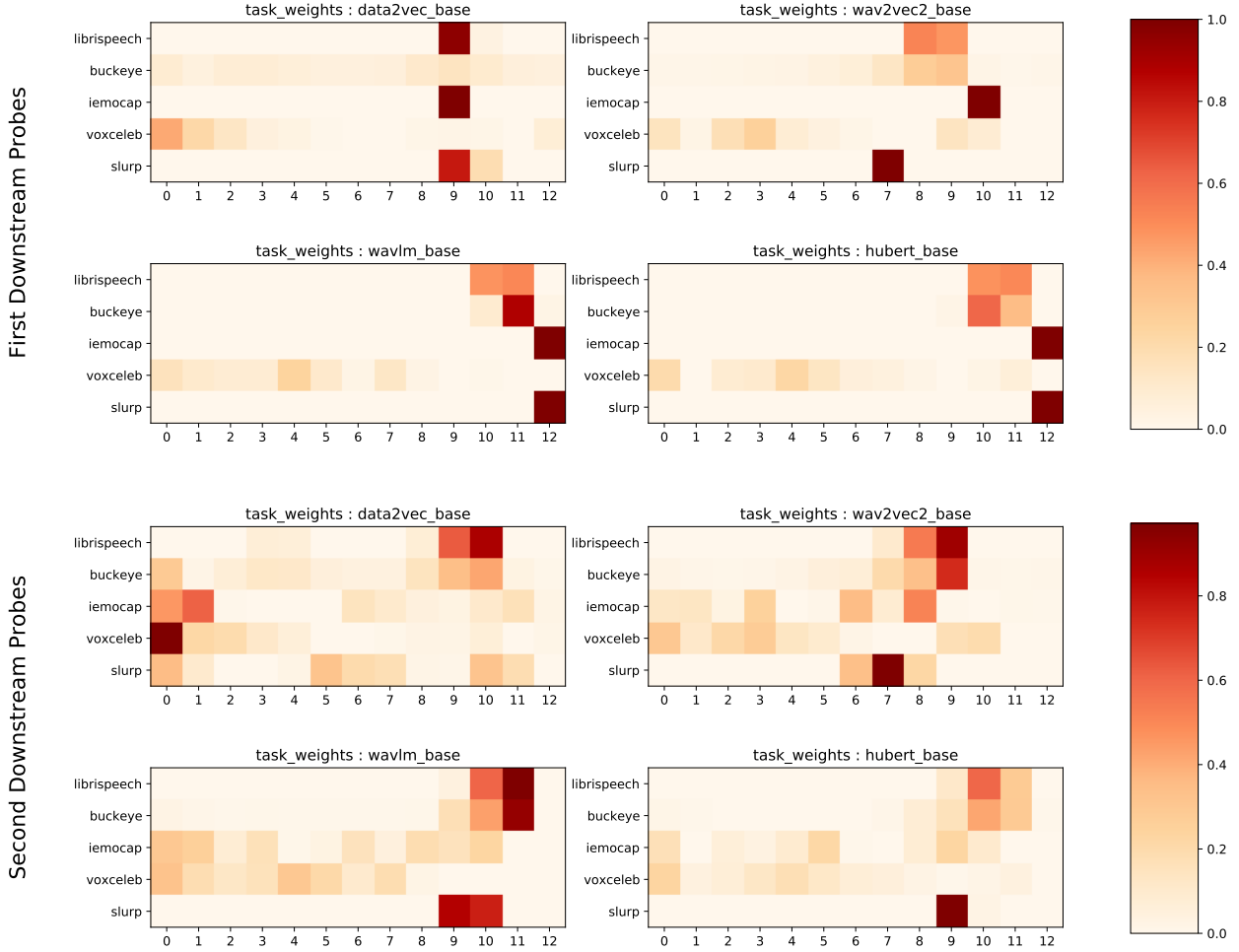


Fig. 2. Values of the layer weights learned during fine-tuning for all “Base” models on the considered tasks. The values on every row sum to 1. The weights obtained with the second downstream probes (bottom part of the figure) are shifted to lower-level layers compared to the first probes ones (top part).

This multi-level extraction may be behind the substantial increase in performance for both intent classification and emotion recognition.

We test this conjecture for emotion recognition with another experiment where one downstream probe is learned using fixed weights obtained with the other one. These results are reported in Table V. Precisely, in this experiment, we fix the weights during the downstream training, with the ones obtained either during the first or second probing. In our set of experiments, for every SSL encoder  $\phi$ , we learn the parameters of a downstream probing head DS and a set of weights for the layers representations  $W$ . In Table V, for every SSL encoder  $\phi$ , every column  $DS(i)/W(j)$  with  $i, j \in 1, 2$ , shows the accuracy after decoding with probing head  $DS(i)$  but with fixed weights  $W(j)$  corresponding to the ones learned initially with  $DS(j)$ . The results show that, while the larger capacity probing head still performs better than the low capacity ones with their considered weightings, a reasonable part of the performance increase is imputable to the change in the level of features used. With the same ECAPA-TDNN decoder, using

TABLE V  
RESULTS OF EXPERIMENTS ON EMOTION RECOGNITION WITH FIXED LAYER-WEIGHTS. THE RESULT IN COLUMN  $DS(i)/W(j)$  IS THE ONE OBTAINED LEARNING THE DOWNSTREAM HEAD OF THE  $i$ -th SET WITH FIXED WEIGHTS CORRESPONDING TO THE ONES LEARNED ORIGINALLY WITH THE  $j$ -th PROBING HEAD. THE DIFFERENCE BETWEEN COLUMN 3 AND 4 SHOWS THAT THE EXPLOITATION OF MULTI-LEVEL FEATURES PLAYS A ROLE IN THE BETTER PERFORMANCE OF DS2.

SSL Model / Head/ Weights	DS1/W1	DS1/W2	DS2/W1	DS2/W2
Data2Vec Base	63	63	62.6	<b>72.1</b>
Data2Vec Large	64	63.9	67.9	<b>71.3</b>
WavLM Base	67.8	67.9	71.6	<b>72.5</b>
WavLM Large	75.3	75.3	72.2	<b>77.6</b>

multi-level features improves the performance from 68.6 to 73.3 mean accuracy on the 4 SSL encoders considered in this experiment. Another interesting observation is that the first downstream head, time-pooling followed with a linear decoder, is not able to better exploit multi-level features, with very similar performances between the two weightings.

We conclude that probing with larger capacity decoders

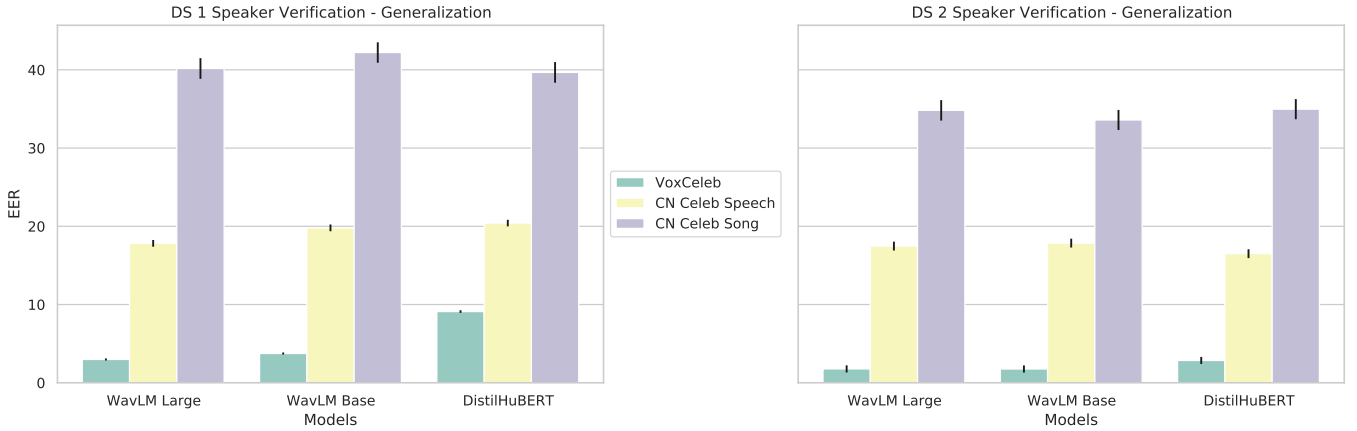


Fig. 3. Generalization performances for automatic speaker verification. CN-Celeb Speech and CN-Celeb Song performances are provided in a zero-shot generalization setting and are not included in the training set. Random performance is at 50 EER, and is not shown for better visualization. Larger probing heads, here ECAPA-TDNN, shown in the right plot, generalize better to out-of-distribution testing samples.

should be preferred if there is a need to exploit multi-level features, as this allows for increased performance. We will show in the next section that it also has an impact on generalization on out-of-domain samples.

### C. Generalization Abilities

A major argument for using low-capacity decoders is that they may allow for better generalization. Indeed, the pre-trained representations are learned on massive amounts of data, with a potential higher data heterogeneity, while the decoding head is learned on small annotated datasets with an expected overfitting hazard. Furthermore, multiple studies have examined and shown the generalization robustness of self-supervised representations [?], [10], which emphasizes, even more, the need to keep this asset. This section aims to show that the models learned with larger capacity decoders are able to generalize as well and even better than their smaller-decoders counterparts, by showing that the performance gains obtained with larger decoders transfer to Out-Of-Domain (OOD) testing samples. Within this scope, we consider the final models obtained with different capacity decoding heads on the considered tasks and test their accuracies on OOD samples, coming from other datasets but having similar downstream classes. This actually enables direct zero-shot generalization performance assessment. Two reasons make two tasks, emotion recognition and speaker verification, relevant for these experiments. First, for both these two tasks, a larger-capacity probing head leads to significantly lower error rates, and we want to test how much this gain is resilient to OOD testing. Second, zero-shot testing requires OOD samples sharing the same labels as the training in-domain set. For ER, several other datasets share, at least partly, the same labels as IEMOCAP [43]. While speaker verification models trained with VoxCeleb [29] output a binary label indicating whether two samples come from the same speaker or not, and thus can be tested on any other ASV benchmark, including OOD non-English utterances.

**Emotion recognition.** To test the generalization abilities of models learned with different decoders, and after training with IEMOCAP as described in Section II, we test the models in a zero-shot fashion, without further fine-tuning, on two datasets: CREMA-D [43] and ASVP-ESD [44]. CREMA-D is a data set of 7,442 original clips from 91 English-speaking actors reading sentences using one of six different emotions (*Anger*, *Disgust*, *Fear*, *Happiness*, *Neutrality*, and *Sadness*). ASVP-ESD is a multi-authentic emotional corpus sourced from movies, Youtube channels, and real-life human interactions in natural settings, without any language limitations. The corpus comprises 5,146 samples, with 60% consisting of non-speech emotional sounds and 40% comprising speech utterances. For both datasets, only speech elements with labels overlapping with the four IEMOCAP ones (*Angry*, *Happy*, *Neutral*, *Sad*) are considered. For these two corpora, the testing sets are of reduced sizes. So to increase the significance of the reported results, and since the train sets are not used for training, all the splits (train and test) are used for testing. For ASVP-ESD, and to further enforce OOD testing, English samples are removed.

**Automatic speaker verification.** For speaker verification, the generalization abilities of the models learned on VoxCeleb1, are tested for two out-of-domain scenarios, also in a zero-shot transfer setting. For this, The CN-Celeb dataset [45], a comprehensive collection of speaker recognition data, is used. It encompasses over 130,000 utterances from 1,000 Chinese celebrities, spanning 11 diverse genres (interviews, movies, songs...). To further highlight generalization ability, we divide CN-Celeb testing couples into ones that include one singing voice element, and once with only spoken utterances, leading to two generalization testing sets: “CN Celeb Speech” and “CN Celeb Song”. The second split is even more challenging in our case, as no singing voice is included in VoxCeleb.

**Discussion.** Figures 3 and 4 show the results of these experiments for models built on certain considered SSL encoders. We can note, first, the expected considerable performance loss on the OOD samples, and especially the loss when changing the ER language with ASVP-ESD or testing on singing voice

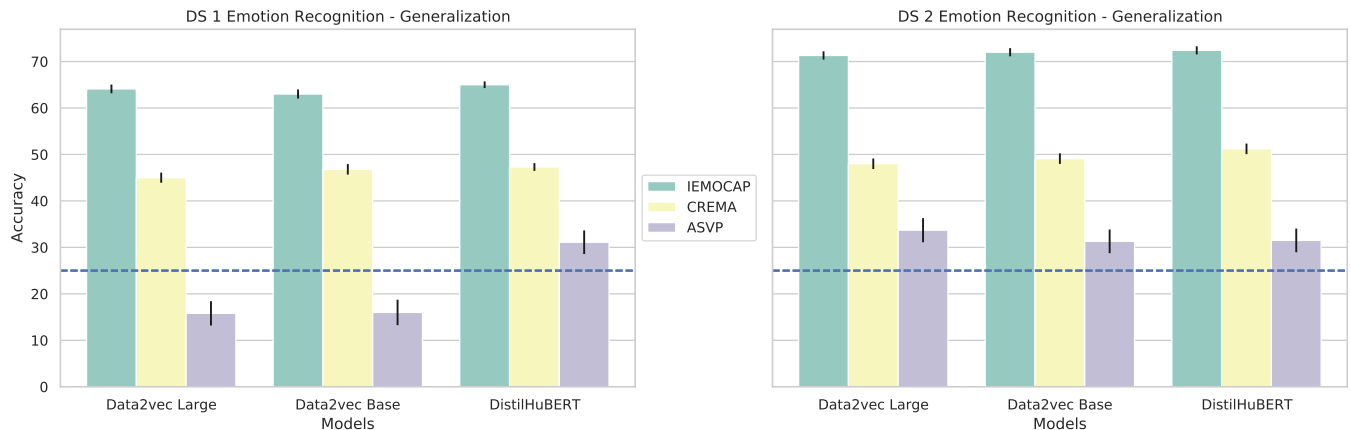


Fig. 4. Generalization performances for emotion recognition. CREMA-D and ASVP-ESD performance is tested in a zero-shot setting. The dashed blue line represents the random accuracy level. Larger probing heads, here ECAPA-TDNN, shown in the right plot, generalize better to out-of-distribution testing samples.

speaker verification with “CN Celeb Song”. For both tasks, as stated in previous sections, in domain performance, *i.e.* performance on the test sets of the downstream training datasets, obtained with the second set of larger probing heads are higher than those with SUPERB limited-capacity probes. The two figures further show that this performance gap stands for zero-shot generalization. Concerning emotion recognition, the mean accuracy on the three considered models reaches 49.43 and 32.17 respectively on CREMA-D and ASVP-SED with the ECAPA-TDNN probing head compared with 46.37 and 20.97 with the time-pooling followed with a linear decoder. For speaker verification, enhancing the probing head drives the Equal Error Rate on the “CN Celeb Speech” from 19.34 to 17.27, while it goes from 40.68 to 34.46 on “CN Celeb Song”. In subsection IV-B, we hypothesized that ER models with the first downstream probes may be using linguistic information since only high-level layers were used. The big drop in performance on ASVP-ESD of Data2Vec “Base” and “Large” models goes in that direction. Changing the language of the inputs leads to catastrophic performance drops. This is not the case for DistilHuBERT as the model only contains three layers. These experiments show that the gain in performance is not only relevant to in-domain data, but models built on top of frozen SSL encoders reach better out-of-domain zero-shot accuracies with larger-capacity probing heads.

## V. CONCLUSION

It is crucial to improve the way the speech community currently benchmarks widely used self-supervised representations. This is important, first because better benchmarks allow SSL users to select properly the models they need for their downstream tasks of interest. Second, it offers the SSL model developers insightful evaluations shaping the training process and decisions. In this work, we have shown, by varying the downstream architectures, that the ranking and relative performances of popular self-supervised models heavily depend on the choice of the probing heads. While the community has previously chosen to evaluate the learned representations with

limited-capacity decoders, we have revealed, as an additional contribution, that larger-capacity decoders should be preferred in various scenarios. This is motivated by better performances, a reduced performance gap between “Base” and “Large” encoders leading to high (performance/inference costs) ratios, better multi-level feature exploitation, and better out-of-distribution generalization. We hope this diagnosis will support the community in designing new benchmarking approaches and encourage submissions to the SUPERB “Constrained” track described in the introduction or propose new probing heads in the dedicated benchmark section within the Speech-Brain Library.

## REFERENCES

- [1] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Neurips*, 2020.
- [2] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhota, R. Salakhutdinov, and A. Mohamed, “Hubert: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 29, oct 2021.
- [3] S. Zaiem, T. Parcollet, S. Essid, and A. Heba, “Pretext tasks selection for multitask self-supervised audio representation learning,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1439–1453, 2022.
- [4] M. Ravanelli, J. Zhong, S. Pascual, P. Swietojanski, J. Monteiro, J. Trmal, and Y. Bengio, “Multi-task self-supervised learning for robust speech recognition,” 2020.
- [5] A. Saeed, D. Grangier, and N. Zeghidour, “Contrastive learning of general-purpose audio representations,” 2020.
- [6] S. Zaiem, T. Parcollet, and S. Essid, “Automatic data augmentation selection and parametrization in contrastive self-supervised speech representation learning,” *Interspeech*, 2022.
- [7] J. Zhao and W.-Q. Zhang, “Improving automatic speech recognition performance for low-resource languages with self-supervised models,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1227–1241, 2022.
- [8] A.-R. Mohamed, H. yi Lee, L. Borgholt, J. D. Havtorn, J. Edin, C. Igel, K. Kirchhoff, S.-W. Li, K. Livescu, L. Maaløe, T. N. Sainath, and S. Watanabe, “Self-supervised speech representation learning: A review,” *IEEE JSTSP*, 2022.
- [9] S. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhota, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, “Superb: Speech processing universal performance benchmark,” 2021.

- [10] T.-h. Feng, A. Dong, C.-F. Yeh, S.-w. Yang, T.-Q. Lin, J. Shi, K.-W. Chang, Z. Huang, H. Wu, X. Chang, S. Watanabe, A. Mohamed, S.-W. Li, and H.-y. Lee, "Superb @ slt 2022: Challenge on generalization and efficiency of self-supervised speech representation learning," 01 2023, pp. 1096–1103.
- [11] S. Evain, H. Nguyen, H. Le, M. Z. Boito, S. Mdahaffar, S. Alisamir, Z. Tong, N. Tomashenko, M. Dinarelli, T. Parcollet, A. Allauzen, Y. Esteve, B. Lecouteux, F. Portet, S. Rossato, F. Ringeval, D. Schwab, and L. Besacier, "Lebenchmark: A reproducible framework for assessing self-supervised representation learning from speech," 2021.
- [12] H.-S. Tsai, H.-J. Chang, W.-C. Huang, Z. Huang, K. Lakhotia, S.-w. Yang, S. Dong, A. Liu, C.-I. Lai, J. Shi, X. Chang, P. Hall, H.-J. Chen, S.-W. Li, S. Watanabe, A. Mohamed, and H.-y. Lee, "SUPERB-SG: Enhanced speech processing universal PERFORMANCE benchmark for semantic and generative capabilities." Dublin, Ireland: ACL, 2022.
- [13] 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," *arXiv preprint arXiv:2303.06740*, 2023.
- [14] Y. Dubois, T. Hashimoto, S. Ermon, and P. Liang, "Improving self-supervised learning by characterizing idealized representations," *arXiv preprint arXiv:2209.06235*, 2022.
- [15] T. Schatz, V. Peddinti, F. Bach, A. Jansen, H. Hermansky, and E. Dupoux, "Evaluating speech features with the Minimal-Pair ABX task: Analysis of the classical MFC/PLP pipeline," in *INTERSPEECH 2013*, Lyon, France, Aug. 2013, pp. 1–5.
- [16] R. Algayres, M. S. Zaiem, B. Sagot, and E. Dupoux, "Evaluating the reliability of acoustic speech embeddings," *INTERSPEECH*, 2020.
- [17] Q. Garrido, R. Balestrierio, L. Najman, and Y. Lecun, "Rankme: Assessing the downstream performance of pretrained self-supervised representations by their rank," *arXiv preprint*, 2022.
- [18] S. Zaiem, Y. Kémiche, T. Parcollet, S. Essid, and M. Ravanelli, "Speech self-supervised representation benchmarking: Are we doing it right?" *arXiv preprint arXiv:2306.00452*, 2023.
- [19] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "Speechbrain: A general-purpose speech toolkit," 2021.
- [20] J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu, P. Mazare, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-light: A benchmark for ASR with limited or no supervision," in *ICASSP*, 2020.
- [21] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 (ICASSP)*, 2015, pp. 5206–5210.
- [22] 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, X. Yu, and F. Wei, "WavLM: Large-Scale Self-Supervised Pre-Training for Full Stack Speech Processing," *IEEE JSTSP*, vol. 16, no. 6, pp. 1505–1518, oct 2021.
- [23] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," in *International Conference on Machine Learning*. PMLR, 2022, pp. 1298–1312.
- [24] H.-J. Chang, S.-w. Yang, and H.-y. Lee, "DistilHuBERT: Speech Representation Learning by Layer-wise Distillation of Hidden-unit BERT," *ICASSP*, 2021.
- [25] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "Transformers: State-of-the-art natural language processing," in *EMNLP*, 2020.
- [26] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," *arXiv preprint arXiv:1904.01038*, 2019.
- [27] M. Pitt, K. Johnson, E. Hume, S. Kiesling, and W. Raymond, "The buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability," *Speech Communication*, vol. 45, pp. 89–95, 01 2005.
- [28] 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.
- [29] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: A large-scale speaker identification dataset," *Interspeech 2017*, Aug 2017.
- [30] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, pp. 335–359, 2008.
- [31] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.
- [32] E. Bastianelli, A. Vanzo, P. Swietojanski, and V. Rieser, "Slurp: A spoken language understanding resource package," *EMNLP 2020*.
- [33] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *Interspeech*, 2020.
- [34] W. Han, Z. Zhang, Y. Zhang, J. Yu, C.-C. Chiu, J. Qin, A. Gulati, R. Pang, and Y. Wu, "Contextnet: Improving convolutional neural networks for automatic speech recognition with global context," *Interspeech*, 2020.
- [35] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *(ICASSP)*. IEEE, 2018, pp. 5329–5333.
- [36] B. Desplanques, J. Thienpondt, and K. Demuynck, "Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification," *Interspeech*, 2020.
- [37] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," *Interspeech*, 2019.
- [38] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, 2018.
- [39] Y. Wang, A. Boumadane, and A. Heba, "A fine-tuned wav2vec 2.0/hubert benchmark for speech emotion recognition, speaker verification and spoken language understanding," *arXiv preprint arXiv:2111.02735*, 2021.
- [40] P. Riera, M. Cerdeiro, L. Pepino, and L. Ferrer, "Phone and speaker spatial organization in self-supervised speech representations," 2023.
- [41] A. Pasad, J.-C. Chou, and K. Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 914–921.
- [42] A. Pasad, B. Shi, and K. Livescu, "Comparative layer-wise analysis of self-supervised speech models," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [43] H. Cao, D. G. Cooper, M. K. Keutmann, R. C. Gur, A. Nenkova, and R. Verma, "Crema-d: Crowd-sourced emotional multimodal actors dataset," *IEEE transactions on affective computing*, vol. 5, no. 4, pp. 377–390, 2014.
- [44] T. T. L. DeJoli, Q. He, and W. Xie, "Audio, Speech and Vision Processing Lab Emotional Sound database (ASVP-ESD)," May 2021. [Online]. Available: <https://doi.org/10.5281/zenodo.7132783>
- [45] Y. Fan, J. Kang, L. Li, K. Li, H. Chen, S. Cheng, P. Zhang, Z. Zhou, Y. Cai, and D. Wang, "Cn-celeb: a challenging chinese speaker recognition dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7604–7608.