Interspeech 2024
1-5 September 2024, Kos, Greece

# ML-SUPERB 2.0: Benchmarking Multilingual Speech Models Across Modeling Constraints, Languages, and Datasets

*Jiatong Shi*[1], *Shih-Heng Wang*[2,*], *William Chen*[1,*], *Martijn Bartelds*[3,*], *Vanya Bannihatti Kumar*[1], *Jinchuan Tian*[1], *Xuankai Chang*[1], *Dan Jurafsky*[3], *Karen Livescu*[3,4], *Hung-yi Lee*[2], *Shinji Watanabe*[1]

[1] Carnegie Mellon University, [2] National Taiwan University, [3] Stanford University,
[4] Toyota Technological Institute at Chicago

{jiatongs, swatanab}@cs.cmu.edu

## Abstract

ML-SUPERB evaluates self-supervised learning (SSL) models on the tasks of language identification and automatic speech recognition (ASR). This benchmark treats the models as feature extractors and uses a single shallow downstream model, which can be fine-tuned for a downstream task. However, real-world use cases may require different configurations. This paper presents ML-SUPERB 2.0, which is a new benchmark for evaluating pre-trained SSL and supervised speech models across downstream models, fine-tuning setups, and efficient model adaptation approaches. We find performance improvements over the setup of ML-SUPERB. However, performance depends on the downstream model design. Also, we find large performance differences between languages and datasets, suggesting the need for more targeted approaches to improve multilingual ASR performance.

**Index Terms**: self-supervised learning, efficient fine-tuning, model adaptation, multilingual speech recognition, benchmarks

## 1. Introduction

Modern multilingual speech models have the capacity to model hundreds or, in some cases, over a thousand languages [1–9], enabled by different training objectives, model architectures, and sources of training data. Importantly, the performance of these models is often evaluated using different experimental setups, which limits the extent to which their performance can be reliably compared. Several standardized evaluation setups and benchmarks have been proposed to evaluate the performance of pre-trained multilingual speech models [10–12].

The most comprehensive benchmark in terms of language coverage is the Multilingual Speech Universal PERformance Benchmark (ML-SUPERB) [13], which covers 143 languages and includes multiple downstream tasks: monolingual ASR, multilingual ASR, and language identification (LID). Like the original SUPERB [14], which only considers English speech, ML-SUPERB is set up to evaluate the performance of self-supervised learning (SSL) models. This evaluation is performed by freezing their representations and treating the models as feature extractors. These features are used as input to a lightweight downstream model, which can be fine-tuned for any of the downstream tasks. To minimize the impact of the downstream model on the overall measured performance, a simple two-layer Transformer-based decoder is used. ML-SUPERB was presented as a challenge at ASRU 2023, attracting 12 model submissions and 8 new language submissions [15–25].

Although the design of ML-SUPERB allows for efficient evaluation of multilingual SSL models across a large number of languages, it only considers one fixed downstream model design. This is problematic, as past work has found that the choice of downstream model can affect the rankings of SSL models across downstream tasks [26, 27]. Also, the choice of downstream model designs can be affected by application requirements and users' budgets, which further motivates benchmarking with more flexible constraints.

In this paper, we present ML-SUPERB 2.0, which revisits ML-SUPERB's original design. Specifically, ML-SUPERB 2.0 includes larger-scale downstream models, SSL model fine-tuning (including partial fine-tuning strategies), efficient pre-trained model adaptation techniques (adapters [28] and LoRA [29]), and supervised pre-trained models (Whisper [3] and OWSM 3.1 [30]). Also, we enrich ML-SUPERB's evaluation metrics to place greater focus on robustness across languages and describe variation across datasets. All code and data used to develop ML-SUPERB 2.0 are publicly available.[1]

## 2. Investigation Details

ML-SUPERB 2.0 considers a variety of architectural variations, pre-training and fine-tuning approaches, described in the next four sections. We then discuss the changes in the evaluation metrics, which allow us to investigate performance differences across languages and datasets.

### 2.1. Downstream Architectures

Past work has found ASR performance differences between downstream architectures when comparing representations from pre-trained SSL models [26, 31]. These findings motivate a systematic comparison to better understand their impact on ASR performance. Therefore, ML-SUPERB 2.0 considers both CTC-based (CTC) and hybrid CTC/attention-based (CTC-ATT) frameworks as adopted in [26, 32–34], and within each framework, compares three architectures, namely the Transformer [35], Conformer [36], and E-Branchformer [37]. In preliminary experiments, we compared these architectures to others (e.g., bi-LSTMs, transducers), and these three were chosen for their better performance or faster convergence.

### 2.2. Model Fine-Tuning

Fine-tuning is a common practice to adapt pre-trained SSL models to a downstream task. While fine-tuning is effective, it traditionally requires updating all model parameters, which is costly. Partial fine-tuning is an alternative that strikes a balance between training efficiency and performance [38, 39]. ML-SUPERB 2.0 includes fine-tuning for the CTC/CTC-ATT

[*] Equal contribution.

[1] https://github.com/espnet/espnet/tree/master/egs2/ml_superb/asr1

10.21437/Interspeech.2024-2248

frameworks, using either full fine-tuning or partial fine-tuning, which focuses on the bottom, middle, or top layers of the models, while keeping the other layers fixed.

### 2.3. Efficient Model Adaptation

Efficient model adaptation approaches offer a parameter-efficient alternative to full fine-tuning [28, 29, 40, 41]. In particular, the use of adapter models has been found to be competitive with, and sometimes improve upon, full fine-tuning, especially in low-resource settings [39, 42–44]. These adapter models are small neural modules added between layers of a pre-trained model, which enable efficient fine-tuning by only learning the adapter module parameters. ML-SUPERB 2.0 evaluates the performance of adapters using the CTC/CTC-ATT frameworks. Specifically, we insert two adapter layers into each layer of the pre-trained SSL models, leaving the rest of the model unchanged (i.e., following the setup of [28]). ML-SUPERB 2.0 also evaluates Low-Rank Adaptation (LoRA). LoRA freezes the pre-trained SSL models and injects low-dimensional layers to be added to the outputs of the projection matrices within the multi-head attention mechanism.

### 2.4. Supervised Pre-Trained Models

Scaling up supervised models has resulted in ASR performance that is competitive with SSL models on several evaluation datasets [3, 45]. ML-SUPERB 2.0 evaluates two recent supervised models, namely Whisper and OWSM 3.1, to relax the constraint of evaluating SSL models only. We use the CTC framework to evaluate the encoder and the CTC-ATT framework to evaluate both the encoder and decoder of these models. Also, we evaluate the partial fine-tuning setup described in Section 2.2 within the CTC framework and use it exclusively within the CTC-ATT framework to limit the number of tunable parameters on the ML-SUPERB 2.0 dataset.

## 3. Experimental Design

ML-SUPERB 2.0 evaluates both multilingual ASR and LID. The objective is to concurrently predict a language identifier token and transcribe the spoken content. ML-SUPERB 2.0 does not include ML-SUPERB's monolingual ASR track.

### 3.1. General Setup

ML-SUPERB 2.0 updates ML-SUPERB's dataset by correcting annotation mistakes,[2] resulting in $\sim$300 hours (85 hours for validation and test sets) drawn from 142 languages across 15 datasets. Some languages occur in more than one dataset. A 1-hour subset was drawn for each language-dataset pair, and the 1-hour subsets were combined to obtain the training dataset. Similarly, 10-minute subsets were drawn for each language-dataset pair, and these serve as the development and test datasets. A subset of 20 languages is reserved for few-shot (FS) learning experiments, whereas the normal experiments refer to other 122 languages. In the FS setting, five randomly selected utterances per language are used for training, while the 10-minute subsets for those languages are used for development and testing.

All experiments are performed using ESPnet [46] with SSL models support from S3PRL [14]. Among the SSL models available, we evaluate XLS-R [1] and MMS [2] due to their superior performance on ML-SUPERB.[3] As in ML-SUPERB, we compute a weighted sum of the layers of the SSL models and the encoder of the supervised models, and use it as input to the downstream models. This is applied to each of our experiments.

In line with the spirit of ML-SUPERB, ML-SUPERB 2.0 limits the number of tunable parameters to 100 million for each evaluated configuration. This constraint ensures that large-scale models can be evaluated across a diverse range of computing environments, improving the accessibility and practicality of ML-SUPERB 2.0.

### 3.2. Downstream Architectures

When evaluating the different architectures within the CTC and CTC-ATT frameworks, we base our hyperparameter selection on prior research [32–34]. In particular, we keep the number of parameters of the downstream models below 100 million and tune only the learning rates. For the CTC framework, the layer configurations are as follows: 24 layers for the Transformer-based model, 14 for the Conformer-based model, and 12 for the E-Branchformer-based model. For the CTC-ATT models' encoders, we use 15 layers for the Transformer-based, 8 for the Conformer-based, and 7 for the E-Branchformer-based models. The Conformer-based model has a kernel size of 15, whereas the E-Branchformer's multi-layer perceptron uses a kernel size of 31 and a dimension of 3072. Common configurations across all models include an 8-head multi-head attention module with 512 hidden states and 2048 projection units, a batch size of 8 with gradient accumulation every four steps, a learning rate chosen from the range $[10^{-3}, 10^{-4}, 10^{-5}]$ with 25,000 warm-up steps, and a dropout rate of 0.1. For the decoders, a Transformer decoder with 8 layers is used for all models. For hybrid training, the CTC and attention decoder weights are set to 0.3 and 0.7 respectively.

### 3.3. Model Fine-Tuning

ML-SUPERB 2.0 evaluates fine-tuning approaches using XLS-R and MMS, which both have 24 layers. The partial fine-tuning approach targets layers 1–6 (bottom), 9–14 (middle), or 19–24 (top). This way, the number of updated parameters does not exceed 100 million. Besides partial fine-tuning, we also examine full fine-tuning, which is provided only for comparison.

To explore the impact of different downstream training objectives, we evaluate both the CTC and CTC-ATT frameworks. The CTC framework uses a 2-layer Transformer-encoder as in ML-SUPERB [13]. For the CTC-ATT framework, we adopt a small-scale downstream model from the configuration in [33] to ensure that there are fewer than than 100 million tunable parameters. Specifically, the model consists of a 2-layer Transformer-based encoder and a 4-layer Transformer-based decoder. Each encoder block has an 8-head multi-head attention module with 256 hidden states and 1024 projection units, and each decoder block contains a 4-head multi-head attention module with 256 hidden states and 2048 linear projection units. The other hyper-parameters are similar to those used for the experiments comparing downstream architectures.

### 3.4. Efficient Model Adaptation

We evaluate the use of adapters and LoRA within both frameworks and follow the setup described in Section 3.2. The con-

[2] We removed Highland Puebla Nahuatl from the Mexican endangered languages corpus and Norwegian from the NST corpus because of their mismatched annotations, and corrected the language label for VoxPopuli Italian.

[3] We use model variants with 24 layers and $\sim$300 million parameters.

figuration of the adapter models and LoRA follow previous work [44]. Specifically, the adapter layers have a dimension of 64, and we set the LoRA rank and its constant scaling factor $\alpha$ to 16. The LoRA module is used across all query and key vectors within the multi-head attention module of the pre-trained SSL models. To accommodate the additional parameters introduced by the adaptation layers, we reduce the number of layers in the encoder of the downstream models by one.

### 3.5. Supervised Pre-Trained Models

ML-SUPERB 2.0 evaluates the medium-sized variants of Whisper and OWSM 3.1, since these are closest in size to the evaluated XLS-R and MMS models.[4] We include two experimental setups using these models, namely one using only their pre-trained encoder within the CTC framework, and another that evaluates both the pre-trained encoder and decoder within the CTC-ATT framework. For the CTC framework, ML-SUPERB 2.0 investigates the performance of both the frozen pre-trained encoder using a Transformer-based downstream model and partial fine-tuning of the pre-trained encoder. The experimental setup is similar to that for the CTC framework described in Sections 3.2 and 3.3, with the exception of fine-tuning only the top layers of the encoder (i.e., layers 19-24) to limit the number of updated parameters to 100 million. In the CTC-ATT framework, we do not add additional downstream models. The encoder remains frozen and we also use the same settings (i.e., medium-sized model variant) as in the CTC framework. Moreover, fine-tuning only targets the top layers of the decoder (i.e., layers 19-24).

### 3.6. Evaluation

For each configuration of the benchmark, ML-SUPERB 2.0 computes the LID accuracy and character error rates (CER) on the test dataset. Specifically, we first compute a per-language CER as the macro-average of CERs across all of the (one or more) datasets per language. We then compute the macro-average of the per-language CERs and the standard deviation (SD) of the language-specific CERs. We report these for both the normal and few-shot (FS) settings. The LID accuracy scores are only reported for the normal setting. Inspired by past work on fairness in machine learning [47], we also report the worst-performing language (WL), i.e. the one with the highest CER in the normal setting, for each configuration, in an attempt to encourage research on methods that leave no language "behind". Lastly, we investigate the CER range between multiple datasets in the same language, when available, to separate the effects of domain or acoustic differences. We perform this analysis using the best-performing model and configuration of the benchmark given the CER in the normal setting. We describe the language that shows the highest range in CER among its datasets.

## 4. Results and Discussion

### 4.1. Comparisons Between Models and Settings

**Downstream Architectures**: The results for different downstream architectures are presented in Table 1. The table shows that there is no superior model across all evaluated configurations. However, the E-Branchformer-based models outperform their Transformer-based and Conformer-based counterparts in almost all cases. This result aligns with trends noted in previous work [34], confirming the strong performance of the E-Branchformer model for LID and multilingual ASR.

Table 1: *Results of the downstream architecture experiments, showing the downstream model, number of model parameters (tunable parameters in parentheses), LID accuracy (ACC), aggregated CERs and few-shot CERs (FS) with standard deviations, and CERs for the worst-performing language (WL). T., C., E-B. are abbreviations for Transformer, Conformer, and E-Branchformer. + indicates the use of the CTC-ATT framework. † refers to the original ML-SUPERB setting [13].*

| Models | Method | Param. (M) | ACC | CER | | |
|---|---|---|---|---|---|---|
| | | | | Normal | FS | WL |
| XLS-R | T.$^\dagger$ | 323.7 ( 6.3) | 90.9 | 24.8 ± 12.1 | 34.4 ± 21.1 | 75.1 |
| MMS | T.$^\dagger$ | 321.8 ( 6.3) | 90.3 | 24.7 ± 12.3 | **31.0** ± 18.6 | 67.6 |
| XLS-R | T. | 408.5 (91.1) | 93.7 | 20.7 ± 10.8 | 33.3 ± 20.8 | 68.0 |
| | C. | 408.9 (91.5) | 82.3 | 22.9 ± 12.8 | 33.4 ± 20.5 | 86.9 |
| | E-B. | 409.6 (92.2) | 94.1 | 18.2 ± 10.6 | 32.3 ± 20.9 | 69.5 |
| | T.$^+$ | 416.0 (98.6) | 93.6 | 19.2 ± 11.9 | 33.6 ± 21.0 | 76.2 |
| | C.$^+$ | 416.3 (98.9) | 83.7 | 23.9 ± 19.1 | 34.8 ± 22.6 | 102.9 |
| | E-B.$^+$ | 417.1 (99.7) | 94.7 | 16.9 ± 10.7 | 32.3 ± 21.1 | 63.8 |
| MMS | T. | 406.6 (91.1) | 93.6 | 21.0 ± 11.2 | 31.7 ± 19.3 | 67.4 |
| | C. | 407.0 (91.5) | 85.3 | 22.7 ± 14.2 | 31.7 ± 17.7 | 94.6 |
| | E-B. | 407.7 (92.2) | 93.0 | 20.4 ± 10.6 | **31.0** ± 19.1 | **61.5** |
| | T.$^+$ | 414.1 (98.6) | 94.3 | 18.8 ± 11.8 | 31.9 ± 19.0 | 73.1 |
| | C.$^+$ | 414.4 (98.9) | 84.0 | 23.8 ± 16.7 | 33.6 ± 18.5 | 106.1 |
| | E-B.$^+$ | 415.2 (99.7) | **95.2** | **16.6** ± 11.8 | 32.6 ± 20.4 | 69.8 |

Table 2: *Results of the fine-tuning experiments, showing the method, number of model parameters (tunable parameters in parentheses), LID accuracy (ACC), aggregated CERs and few-shot CERs (FS) with standard deviations, and CERs for the worst-performing language (WL). + indicate the use of the CTC-ATT framework. † refers to the original ML-SUPERB setting [13].*

| Models | Method | Param. (M) | ACC | CER | | |
|---|---|---|---|---|---|---|
| | | | | Normal | FS | WL |
| XLS-R | -$^\dagger$ | 323.7 ( 6.3) | 90.9 | 24.8 ± 12.1 | 34.4 ± 21.1 | 75.1 |
| MMS | -$^\dagger$ | 321.8 ( 6.3) | 90.3 | 24.7 ± 12.3 | 31.0 ± 18.6 | 67.6 |
| XLS-R | 1-6 | 323.7 ( 90.3) | 91.7 | 20.5 ± 12.8 | 29.4 ± 17.8 | 74.0 |
| | 9-14 | 323.7 ( 90.3) | 93.0 | 18.5 ± 12.8 | 31.3 ± 21.3 | 73.2 |
| | 19-24 | 323.7 ( 90.3) | 91.4 | 22.0 ± 13.2 | 31.8 ± 20.8 | 74.8 |
| | 1-24 | 323.7 (323.7) | 94.3 | 15.8 ± 12.4 | 28.6 ± 20.2 | 70.2 |
| | 1-6$^+$ | 333.4 ( 99.9) | 84.0 | 30.5 ± 22.8 | 35.4 ± 18.1 | 119.1 |
| | 9-14$^+$ | 333.4 ( 99.9) | 93.2 | 22.7 ± 18.3 | 32.2 ± 18.7 | 96.0 |
| | 19-24$^+$ | 333.4 ( 99.9) | 89.8 | 25.6 ± 19.5 | 32.2 ± 18.4 | 101.5 |
| | 1-24$^+$ | 333.4 (333.4) | 94.1 | 16.8 ± 14.3 | 29.5 ± 17.2 | 79.0 |
| MMS | 1-6 | 321.8 ( 90.8) | 93.8 | 18.8 ± 12.0 | 31.0 ± 20.8 | 75.6 |
| | 9-14 | 321.8 ( 90.8) | 95.6 | **15.5** ± 10.3 | **27.7** ± 16.7 | **62.7** |
| | 19-24 | 321.8 ( 90.8) | 93.4 | 19.4 ± 14.6 | 28.5 ± 17.8 | 96.2 |
| | 1-24 | 321.8 (321.8) | 87.7 | 27.4 ± 13.6 | 31.7 ± 18.8 | 80.5 |
| | 1-6$^+$ | 331.4 (100.5) | 93.6 | 25.4 ± 16.4 | 35.9 ± 19.6 | 91.2 |
| | 9-14$^+$ | 331.4 (100.5) | **95.7** | 17.6 ± 14.6 | 28.9 ± 16.8 | 89.5 |
| | 19-24$^+$ | 331.4 (100.5) | 92.1 | 23.2 ± 21.6 | 28.5 ± 17.5 | 119.7 |
| | 1-24$^+$ | 331.4 (331.4) | 95.5 | 15.9 ± 15.0 | 30.2 ± 20.7 | 81.6 |

When comparing the CTC and CTC-ATT frameworks, we find that CTC performs slightly better in the few-shot setting, while CTC-ATT (i.e, rows with a plus) is stronger in the normal setting. The findings suggest that the CTC framework might have better generalization capabilities when limited amounts of data are available. Comparing these results to the shallow-downstream baseline from ML-SUPERB (i.e., first two rows), we find an improvement in LID and ASR performance in the normal setting. However, the shallow-downstream baseline, based on MMS, still performs competitively in the few-shot setting. With roughly 6 million tunable parameters, the baseline's performance echos the insight from the 2023 ML-SUPERB challenge [15]: scaling up models does not necessarily translate to improved performance on multilingual speech tasks.

In sum, our results reinforce findings in past work [26] that

[4] The Whisper and OWSM 3.1 model variants have 769 and 1017 million parameters, respectively.

Table 3: *Results of the efficient model adaptation experiments, showing the method, number of model parameters (tunable parameters in parentheses), LID accuracy (ACC), aggregated CERs and few-shot CERs (FS) with standard deviations, and CERs for the worst-performing language (WL). + indicate the use of the CTC-ATT framework. † refers to the original ML-SUPERB setting [13].*

| Models | Method | Param. (M) | ACC | CER | | |
|---|---|---|---|---|---|---|
| | | | | Normal | FS | WL |
| XLS-R | -$^\dagger$ | 323.7 ( 6.3) | 90.9 | 24.8 ± 12.1 | 34.4 ± 21.1 | 75.1 |
| MMS | -$^\dagger$ | 321.8 ( 6.3) | 90.3 | 24.7 ± 12.3 | **31.0** ± 18.6 | 67.6 |
| XLS-R | LoRA | 410.1 ( 92.7) | **94.4** | 20.3 ± 10.7 | 33.2 ± 21.2 | **63.0** |
| | Adapter | 411.7 ( 94.3) | 94.2 | 20.6 ± 10.8 | 33.7 ± 20.8 | 67.3 |
| | LoRA$^+$ | 415.8 ( 98.4) | 93.8 | 19.1 ± 11.9 | 33.7 ± 20.7 | 69.7 |
| | Adapter$^+$ | 417.4 (100.0) | 93.4 | 19.5 ± 11.7 | 33.3 ± 20.8 | 72.3 |
| MMS | LoRA | 408.2 ( 92.7) | 93.5 | 21.3 ± 10.9 | 31.5 ± 18.3 | 65.8 |
| | Adapter | 409.8 ( 94.3) | 91.7 | 24.5 ± 11.0 | 35.5 ± 18.9 | 70.6 |
| | LoRA$^+$ | 413.7 ( 98.4) | 94.2 | **18.7** ± 11.5 | 32.6 ± 20.0 | 68.0 |
| | Adapter$^+$ | 415.5 (100.0) | 92.3 | 21.9 ± 12.2 | 35.7 ± 19.5 | 77.2 |

Table 4: *Results of the supervised model experiments, showing whether fine-tuning (FT) is performed, number of model parameters (tunable parameters in parentheses), LID accuracy (ACC), aggregated CERs and few-shot CERs (FS) with standard deviations, and CERs for the worst-performing language (WL). Asterisks indicate that only the encoder is used. † refers to the original ML-SUPERB setting [13].*

| Models | FT | Param. (M) | ACC | CER | | |
|---|---|---|---|---|---|---|
| | | | | Normal | FS | WL |
| XLS-R | ✗$^\dagger$ | 323.7 ( 6.3) | 90.9 | 24.8 ± 12.1 | 34.4 ± 21.1 | 75.1 |
| MMS | ✗$^\dagger$ | 321.8 ( 6.3) | 90.3 | 24.7 ± 12.3 | 31.0 ± 18.6 | **67.6** |
| Whisper | ✗* | 515.8 ( 91.1) | **91.7** | **21.0** ± 12.5 | **27.4** ± 13.3 | 82.9 |
| | ✓* | 431.0 ( 90.7) | 83.9 | 26.8 ± 15.0 | 29.6 ± 13.5 | 93.5 |
| | ✓ | 762.3 ( 84.4) | 85.5 | 25.6 ± 19.4 | 35.0 ± 17.5 | 107.2 |
| OWSM | ✗* | 671.2 ( 88.4) | 77.8 | 27.8 ± 22.6 | 31.7 ± 17.3 | 99.9 |
| | ✓* | 612.1 ( 88.4) | 71.0 | 24.9 ± 14.9 | 31.5 ± 16.9 | 99.7 |
| | ✓ | 1016.9 (100.8) | 80.5 | 40.0 ± 41.8 | 40.0 ± 24.9 | 337.6 |

pre-trained SSL model rankings for ASR vary with the choice of downstream architecture.

**Model Fine-tuning**: The model fine-tuning results are presented in Table 2. These results suggest that fine-tuning of the middle layers (i.e., layers 9–14) is more effective across the evaluated SSL models and training frameworks than fine-tuning the bottom or top layers. While full fine-tuning mostly outperforms partial fine-tuning in the normal setting (it also has the lowest mean CER on the worst-performing language in most cases), this is not the case in the FS setting. For instance, full fine-tuning of MMS leads to a higher mean CER compared to fine-tuning the middle layers in the FS setting. This suggests that the choice of fine-tuning strategy is crucial and warrants further exploration within the context of the benchmark.

**Efficient Model Adaptation**: The efficient model adaptation results, detailed in Table 3, also do not reveal a single best model across the evaluated configurations. However, LoRA outperforms adapters across SSL models in the normal setting, indicating it is the preferred option within the setup of the benchmark. When comparing frameworks, the results generally align with those from the downstream analysis (Table 1). We find a difference when looking at the LID task, where XLS-R with LoRA adaptation outperforms MMS within the CTC framework, while MMS achieves better performance within the CTC-ATT framework. This suggests that the choice of framework and adaptation method can impact the performance, depending on the task and the SSL model used.

**Supervised Pre-Trained Models**: The experiments with supervised pre-trained models are shown in Table 4. The results indicate that using only the pre-trained encoder from supervised models leads to better ASR performance than using models with the original decoder. The performance differences might stem from challenges in partial fine-tuning of the decoder, or from the potential biases from large-scale supervised training in major languages. Also, we find that supervised pre-trained models do not consistently outperform the SSL-based models across the evaluated configurations, which aligns with results reported in previous work [45]. While this work does not conduct a deeper analysis into the optimal utilization of supervised pre-trained models, it highlights this area as a promising direction for future research within the ML-SUPERB 2.0 benchmark.

### 4.2. Variation Across Languages and Datasets

To investigate the impact of different languages on the benchmark performance, we report a standard deviation for each reported CER. We find large standard deviations in both the normal and few-shot settings, indicating that there is substantial variation among the language-specific CERs. The CER of the worst-performing language, which we found to be Lao or Min Nan Chinese in most cases, also highlights the large impact of language differences, since it is substantially higher than the mean CER in the normal and few-shot settings.

When investigating performance differences between datasets within a single language, we find large differences as well. For the best-performing model and configuration of ML-SUPERB 2.0, which involves fine-tuning the middle layers of MMS within the CTC framework, the largest differences in CER are among the datasets of Urdu. Specifically, we find that the CER of Urdu from Common Voice [48] is 21.8%, whereas it is 56.9% on data from Fleurs [49]. Note also that Urdu has the largest performance difference between its datasets in many of the other evaluated configurations.

These results motivate future work on creating truly multilingual model representations, which can transfer to a broad range of languages and domains.

## 5. Conclusion

We introduced ML-SUPERB 2.0, an updated benchmark for multilingual speech pre-trained models, which builds upon and extends ML-SUPERB. By relaxing many of ML-SUPERB's constraints, ML-SUPERB 2.0 opens up new avenues for research, offering a broader scope for exploration within the benchmark's setup. We investigated four primary extensions to ML-SUPERB, namely the use of larger-scale downstream models, model fine-tuning, efficient model adaptation, and the incorporation of supervised pre-trained models. Furthermore, we enhanced the evaluation metrics of ML-SUPERB to better track robustness across languages, and described dataset variation using the benchmark's best-performing model and configuration.

While each of the four extensions has shown improvements over the models in the original ML-SUPERB, model fine-tuning achieves the best performance on both LID and multilingual ASR. However, the large deviations across languages and the substantially higher CER for the worst-performing languages suggest that tailored or language-specific approaches might be essential to reduce performance variability and improve model efficacy in multilingual speech processing.

# 6. Acknowledgements

This work used the Bridges2 system at PSC and Delta system at NCSA through allocations CIS210014 and IRI120008P from the ACCESS program, supported by NSF grants #2138259, #2138286, #2138307, #2137603, and #2138296.

# 7. References

[1] A. Babu *et al.*, "XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale," in *Proc. Interspeech*, 2022.

[2] V. Pratap *et al.*, "Scaling speech technology to 1,000+ languages," *Journal of Machine Learning Research*, vol. 25, no. 97, 2024.

[3] A. Radford *et al.*, "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*, 2023.

[4] Y. Peng *et al.*, "Reproducing Whisper-style training using an open-source toolkit and publicly available data," in *Proc. ASRU*, 2023.

[5] W. Hou *et al.*, "Large-Scale End-to-End Multilingual Speech Recognition and Language Identification with Multi-Task Learning," in *Proc. Interspeech*, 2020.

[6] V. Pratap *et al.*, "Massively Multilingual ASR: 50 Languages, 1 Model, 1 Billion Parameters," in *Proc. Interspeech*, 2020.

[7] X. Li *et al.*, "ASR2K: Speech Recognition for Around 2000 Languages without Audio," in *Proc. Interspeech*, 2022.

[8] Y. Zhang *et al.*, "Google USM: Scaling automatic speech recognition beyond 100 languages," *arXiv preprint arXiv:2303.01037*, 2023.

[9] W. Chen *et al.*, "Improving massively multilingual ASR with auxiliary CTC objectives," in *Proc. ICASSP*, 2023.

[10] A. Conneau *et al.*, "XTREME-S: Evaluating Cross-lingual Speech Representations," in *Proc. Interspeech*, 2022.

[11] L. Della Libera *et al.*, "CL-MASR: A continual learning benchmark for multilingual ASR," *arXiv preprint arXiv:2310.16931*, 2023.

[12] T. Javed *et al.*, "Indicsuperb: A speech processing universal performance benchmark for indian languages," in *Proc. AAAI*, vol. 37, 2023.

[13] J. Shi *et al.*, "ML-SUPERB: Multilingual Speech Universal PERformance Benchmark," in *Proc. Interspeech*, 2023.

[14] S.-W. Yang *et al.*, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech*, 2021.

[15] J. Shi *et al.*, "Findings of the 2023 ML-SUPERB challenge: Pre-training and evaluation over more languages and beyond," in *Proc. ASRU*, 2023.

[16] Y.-H. Chou *et al.*, "Evaluating self-supervised speech models on a Taiwanese Hokkien corpus," in *Proc. ASRU*, 2023.

[17] S. Sakti and B. A. Titalim, "Leveraging the multilingual indonesian ethnic languages dataset in self-supervised model for low-resource asr task," in *Proc. ASRU*, 2023.

[18] T. Ogunremi *et al.*, "Ìròyìnspeech: A multi-purpose yorùbá speech corpus," in *Proc. LREC-COLING*, 2024.

[19] C.-C. Chen *et al.*, "Evaluating self-supervised speech representations for indigenous American languages," in *Proc. LREC-COLING*, 2024.

[20] A. Suwanbandit *et al.*, "Thai dialect corpus and transfer-based curriculum learning investigation for dialect automatic speech recognition," in *Proc. Interspeech*, 2023.

[21] S. Sakti and S. Nakamura, "Towards language preservation: Design and collection of graphemically balanced and parallel speech corpora of indonesian ethnic languages," in *Proc. O-COCOSDA/CASLRE*, 2013.

[22] S. Cahyawijaya *et al.*, "NusaCrowd: Open source initiative for Indonesian NLP resources," in *Findings of ACL*, 2023.

[23] T. Srivastava *et al.*, "EFFUSE: Efficient self-supervised feature fusion for E2E ASR in multilingual and low resource scenarios," in *Proc. Interspeech*, 2024.

[24] W. Chen *et al.*, "Joint prediction and denoising for large-scale multilingual self-supervised learning," in *Proc. ASRU*, 2023.

[25] H. Xue *et al.*, "SSHR: Leveraging self-supervised hierarchical representations for multilingual automatic speech recognition," *arXiv preprint arXiv:2309.16937*, 2023.

[26] S. Zaiem *et al.*, "Speech Self-Supervised Representation Benchmarking: Are We Doing it Right?" In *Proc. Interspeech*, 2023.

[27] S. Arora *et al.*, "On the evaluation of speech foundation models for spoken language understanding," in *Proc. ACL*, 2024.

[28] N. Houlsby *et al.*, "Parameter-efficient transfer learning for NLP," in *Proc. ICML*, 2019.

[29] E. J. Hu *et al.*, "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2021.

[30] Y. Peng *et al.*, "OWSM v3. 1: Better and faster open Whisper-style speech models based on E-Branchformer," in *Proc. Interspeech*, 2024.

[31] X. Chang *et al.*, "An exploration of self-supervised pretrained representations for end-to-end speech recognition," in *Proc. ASRU*, 2021.

[32] S. Karita *et al.*, "A comparative study on Transformer vs RNN in speech applications," in *Proc. ASRU*, 2019.

[33] P. Guo *et al.*, "Recent developments on ESPnet toolkit boosted by conformer," in *Proc. ICASSP*, 2021.

[34] Y. Peng *et al.*, "A comparative study on E-Branchformer vs Conformer in speech recognition, translation, and understanding tasks," in *Proc. Interspeech*, 2023.

[35] A. Vaswani *et al.*, "Attention is all you need," *Proc. NeurIPS*, vol. 30, 2017.

[36] A. Gulati *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech*, 2020.

[37] K. Kim *et al.*, "E-branchformer: Branchformer with enhanced merging for speech recognition," in *Proc. SLT*, 2023.

[38] J. Kunze *et al.*, "Transfer learning for speech recognition on a budget," in *Proc. RepL4NLP*, P. Blunsom *et al.*, Eds., 2017.

[39] W. Hou *et al.*, "Exploiting adapters for cross-lingual low-resource speech recognition," *TASLP*, vol. 30, 2021.

[40] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proc. EMNLP*, 2021.

[41] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. ACL*, 2021.

[42] N.-Q. Pham, A. Waibel, and J. Niehues, "Adaptive multilingual speech recognition with pretrained models," in *Proc. Interspeech*, 2022.

[43] B. Thomas, S. Kessler, and S. Karout, "Efficient adapter transfer of self-supervised speech models for automatic speech recognition," in *Proc. ICASSP*, 2022.

[44] Z.-C. Chen *et al.*, "Exploring efficient-tuning methods in self-supervised speech models," in *Proc. SLT*, 2023.

[45] A. Rouditchenko *et al.*, "Comparison of Multilingual Self-Supervised and Weakly-Supervised Speech Pre-Training for Adaptation to Unseen Languages," in *Proc. Interspeech*, 2023.

[46] S. Watanabe *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018.

[47] T. Hashimoto *et al.*, "Fairness without demographics in repeated loss minimization," in *Proc. ICML*, 2018.

[48] R. Ardila *et al.*, "Common voice: A massively-multilingual speech corpus," in *Proc. LREC*, 2020.

[49] A. Conneau *et al.*, "Fleurs: Few-shot learning evaluation of universal representations of speech," in *Proc. SLT*, 2023.