# JOINT UNSUPERVISED AND SUPERVISED TRAINING FOR MULTILINGUAL ASR

*Junwen Bai*, Bo Li, Yu Zhang, Ankur Bapna, Nikhil Siddhartha, Khe Chai Sim, Tara N. Sainath*

Google, USA

{junwen,boboli,ngyuzh,ankurbpn,nikhilsid,khechai,tsainath}@google.com

## ABSTRACT

Self-supervised training has shown promising gains in pretraining models and facilitating the downstream finetuning for speech recognition, like multilingual ASR. Most existing methods adopt a 2-stage scheme where the self-supervised loss is optimized in the first pretraining stage, and the standard supervised finetuning resumes in the second stage. In this paper, we propose an end-to-end (E2E) Joint Unsupervised and Supervised Training (JUST) method to combine the supervised RNN-T loss and the self-supervised contrastive and masked language modeling (MLM) losses. We validate its performance on the public dataset *Multilingual LibriSpeech* (MLS), which includes 8 languages and is extremely imbalanced. On MLS, we explore (1) JUST trained from scratch, and (2) JUST finetuned from a pretrained checkpoint. Experiments show that JUST can consistently outperform other existing state-of-the-art methods, and beat the monolingual baseline by a significant margin, demonstrating JUST's capability of handling low-resource languages in multilingual ASR. Our average WER of all languages outperforms average monolingual baseline by 33.3%, and the state-of-the-art 2-stage XLSR by 32%. On low-resource languages like Polish, our WER is less than half of the monolingual baseline and even beats the supervised transfer learning method which uses external supervision.

***Index Terms—*** joint training, multilingual ASR, self-supervised learning, contrastive learning

## 1. INTRODUCTION

Self-supervised learning is an effective method in unveiling the useful and general latent representations from large-scale unlabeled data. It is often adopted to pretrain a sequence-to-sequence model and facilitate downstream tasks [1, 2, 3]. In speech recognition, recent works have shown successes of the 2-stage pretrain-finetune schemes [4, 5, 6, 7, 8]. Pretrained models can greatly reduce the sample complexity for downstream finetuning. For instance, finetuning wav2vec 2.0 (w2v2) pretrained on 60k hours with only 1h labeled data can outperform most fully supervised models [9].

While self-supervised learning has been successful for sequence modeling, some concerns have also been raised. For example, finetuning a pretrained model is prone to catastrophic forgetting [10, 11]. The model might *forget* the previously learnt knowledge when trained with supervision, particularly when the supervised set is large. Another concern is the pretrained checkpoint selection. The downstream performance varies from one checkpoint to another, and the one pretrained longer may not be the best one. These issues are even more severe in multilingual ASR, since different languages are often heterogeneous and the corpus is often imbalanced. In *Multilingual LibriSpeech* (MLS) [12], English has up to 44k hours while Polish only has 100 hours. Most existing methods tackle multilingual ASR from 2 directions. The first direction is transfer learning from a source multilingual corpus to a target low-resource multilingual dataset. In [13], the work first trains the model on Google's 15-language VoiceSearch (VS) traffic and then uses it to seed the transfer learning on MLS. Even though some languages in MLS are not included in VS, the model can deliver satisfactory WERs on those low-resource languages, demonstrating its generalization capability. However, such transfer learning requires massive supervised source corpora which may not be easily accessible. Another direction is to learn useful representations through pretraining and perform finetuning with supervision, similar to monolingual ASR. [14] explores the unsupervised pretraining using cross-lingual language modeling and [15] investigates the cross-lingual transfer of phoneme features. XLSR [16] builds on w2v2 and pretrains the model on 53 languages using the self-supervised losses. XLSR also stands for the state-of-the-art (SOTA) on MLS dataset.

In this paper, we propose a novel Joint Unsupervised and Supervised Training (JUST) method for multilingual ASR, to reconcile the unsupervised and supervised losses synergistically. JUST includes two self-supervised losses, contrastive loss [9] and MLM loss [1], together with a supervised RNN-T loss [17]. Our model architecture inherits from w2v-bert [7], a novel variation of w2v2. The outputs from w2v-bert are passed to the decoder and produce the RNN-T loss. We explore two types of learning with JUST: 1) JUST trained from scratch, and 2) JUST finetuned from a pretrained checkpoint. We compare these 2 settings with XLSR and other standard baselines. Experiments show that JUST can consistently outperform other SOTA and baselines. For instance, on 8 languages from MLS, JUST improves over XLSR by 30% on average.

## 2. RELATED WORK

Early works adopted joint training to learn robust and transferable representations. In NLP, [18] proposes joint training for machine translation. [19] suggests multiple pretraining objectives for domain-adaptive applications. In speech, PASE [20] jointly solves multiple self-supervised tasks to learn general representations. More recent research found the joint training with both supervised and unsupervised losses can directly optimize the ASR performance. [21] alternatively minimizes an unsupervised masked CPC loss and a supervised CTC loss [22]. This single-stage method is shown to match the performance of the two-stage w2v2 on the Librispeech 100-hours dataset. Similarly, UniSpeech [23] optimizes a combination of phonetic CTC loss and contrastive loss. To further increase the quantizer codebook usage, UniSpeech randomly replaces contextual representations with quantized latent codes. [24] also designs a similar hybrid multitask learning to train acoustic models under low-resource settings, comprising of supervised CTC, attention and self-supervised reconstruction losses. Similarly, [25] combines

*Work done during an internship at Google.

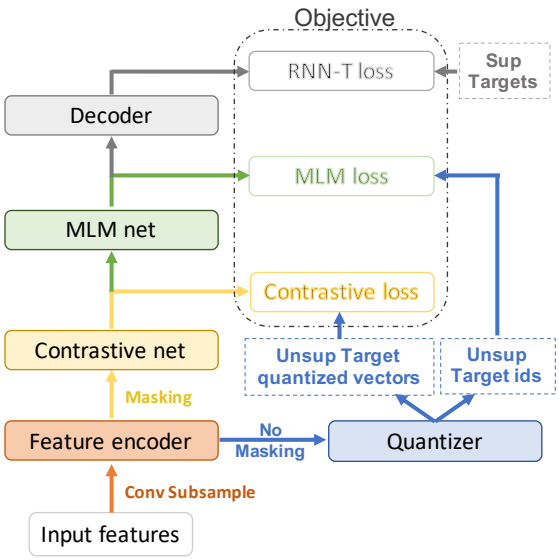


**Fig. 1**: An overview of our JUST framework. Feature encoder, contrastive net, MLM net and decoder are stacked sequentially. The output of each module constitutes a loss in the objective function. Target vectors and ids in the blue boxes are for unsupervised losses. Supervised targets in the grey box are for RNN-T loss.

self- and semi-supervised learning methods for online ASR model. These methods only contain one self-supervised loss in their optimization and often tackle with speech recognition in the phoneme level [23, 24]. JUST incorporates two self-supervised losses (contrastive and MLM losses), and replaces the CTC loss with an RNN-T loss. RNN-T extends CTC with a prediction network to simulate the effect of LM and has been widely adopted in prior multilingual ASR systems [13]. Furthermore, unlike [20, 24] where each of the multiple tasks has its own branch, JUST computes different losses simply using the intermediate outputs from different layers (Fig. 1).

## 3. METHOD

Our JUST framework is comprised of multiple modules for unsupervised and supervised losses. All modules (except for the quantizer) are stacked sequentially and each reads the output from the previous module. We will elaborate on them, along with the losses, in the following sections. Fig. 1 presents an overview.

### 3.1. Feature encoder

The feature encoder converts the original log-mel filter bank features $\{x_i\}_{i=1}^{L}$ to the latent speech representations $\{z_i\}_{i=1}^{T}$ for $T$ time steps. $T$ is smaller than the original length $L$ due to time reduction. Unlike [9] where seven blocks of CNN are used, JUST only has two CNN blocks both with filter size 3×3 and strides (2, 2), the same as [5]. One can also view the feature encoder as a convolutional subsampling, with 4x reduction in the feature dimensionality and sequence length.

### 3.2. Quantizer

JUST adopts a complex quantization mechanism [9]. After the abstraction of the original inputs through feature encoder, the latent representations $\{z_i\}_{i=1}^{T}$ are passed to a quantizer (without any masking). The goal of the quantizer is to "summarize" all the latent speech representations to a finite set (referred to as a codebook) of representative discriminative speech tokens $\{e_j\}_{j=1}^{V}$ where $V$ is the size of the codebook. The codebook in the quantizer stores all these tokens and each latent representation from the feature encoder is mapped to a token index corresponding to a token in the codebook, through Gumbel softmax [26] which enables differentiation of discrete codebook selection. JUST uses a single codebook rather than multiple ones [9]. All the tokens in the codebook are learnable during training. Quantizer module generates target quantized vector (token) $q_i$ and target id (token index) $y_i$ for each $z_i$, where $q_i \in \{e_j\}_{j=1}^{V}$, $y_i \in [1...V]$. To encourage the use of the codebook, [9] introduces the entropy-based diversity loss $\mathcal{L}_d$. We include it in JUST as well.

### 3.3. Contrastive net

The outputs of feature encoder $\{z_i\}_{i=1}^{T}$ are not only used for quantization, but also fed into the contrastive net after masking. For masking, some $z_i$'s are randomly chosen and replaced with random vectors. Contrastive net reads $z_i$ of all time steps (either masked or unmasked) and outputs contrastive context vectors $\{c_i\}_{i=1}^{T}$ for deriving contrastive self-supervised loss. Contrastive net is a stack of Conformer blocks [27], each with multi-headed self-attention, depth-wise convolution and feed-forward layers. To derive the contrastive loss $\mathcal{L}_c$, for anchor $c_i$, we take $q_i$ as the positive sample and $K$ negative samples/distractors $\{\tilde{q}_i\}_{i=1}^{K}$ uniformly sampled from $q_j$ of other masked $z_j$'s in the same utterance:

$$\mathcal{L}_c = -\log \frac{sim(c_i, q_i)}{sim(c_i, q_i) + \sum_{j=1}^{K} sim(c_i, \tilde{q}_j)} \quad (1)$$

where $sim(a, b)$ is the exponential of the cosine similarity between $a$ and $b$.

### 3.4. MLM net

We further boost the contextualized representation learning through a masked prediction task with the quantizer. The inputs of MLM net are $\{c_i\}_{i=1}^{T}$ from contrastive net. Similar to contrastive net, MLM net is also a stack of Conformer blocks. We denote the outputs of MLM net as $\{m_i\}_{i=1}^{T}$, which are high-level context vectors. Each $m_i$ is used for token id prediction through a linear layer. The predicted id $\hat{y}_i \in [1..V]$ is compared with the target token id $y_i$ from the quantizer, by the standard cross-entropy loss $\mathcal{L}_m$.

Together with $\mathcal{L}_d$ and $\mathcal{L}_c$, the unsupervised loss is computed as:

$$\mathcal{L}_u = \mathcal{L}_c + \mathcal{L}_m + \alpha \mathcal{L}_d \quad (2)$$

$\alpha$ is set to 0.1, following [7].

### 3.5. Decoder

The decoder of JUST is a 2-layer RNN Transducer. $\{m_i\}_{i=1}^{T}$ are passed through Swish activation, batch normalization, and finally fed into the decoder. The output vocabulary of the decoder is a unified grapheme set pooled from all the 8 languages in MLS. RNN-T loss is used in this work as the supervised loss, denoted by $\mathcal{L}_s$. Our final objective function is simply the combination of $\mathcal{L}_u$ and $\mathcal{L}_s$:

$$\mathcal{L} = \mathcal{L}_s + \beta \mathcal{L}_u \quad (3)$$

$\beta$ is a trade-off weight. $\mathcal{L}$ is optimized via Adam [28].

| Method | External data | **en** | **de** | **nl** | **fr** | **es** | **it** | **pt** | **pl** | Avg | Avg (w/o **en**) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Monolingual [12] | - | 6.76 | 7.10 | 13.09 | 6.58 | 6.68 | 11.78 | 20.52 | 21.66 | 11.8 | 12.5 |
| + 5-gram LM [12] | - | 5.88 | 6.49 | 12.02 | 5.58 | 6.07 | 10.54 | 19.49 | 20.39 | 10.8 | 11.5 |
| XLSR-53 [16] | Y | - | 7.0 | 10.8 | 7.6 | 6.3 | 10.4 | 14.7 | 17.2 | 10.6 | 10.6 |
| B0 (random init.) [13] | Y | 6.1 | 5.5 | 11.9 | 6.9 | 5.8 | 11.9 | 16.2 | 15.4 | 10.0 | 10.5 |
| B0 (15-language model init.) [13] | Y | 6.6 | 5.0 | 11.1 | 6.1 | 4.7 | 10.1 | 15.5 | 10.9 | 8.8 | 9.1 |
| E3 (15-language model init.) [13] | Y | **5.8** | 4.3 | 9.9 | **4.9** | 4.2 | 8.8 | 15.2 | 10.4 | 7.9 | 8.2 |
| JUST ($\beta = 0$) | N | 6.9 | 5.5 | 10.3 | 6.0 | 4.1 | 9.3 | 9.4 | 11.3 | 7.8 | 8.0 |
| w2v2 Pretrain ($\mathcal{L}_c$+$\alpha\mathcal{L}_d$)+pure Finetune ($\mathcal{L}_s$) | N | 6.8 | 4.7 | 10.3 | 5.8 | 4.1 | 9.9 | 12.6 | 12.1 | 8.3 | 8.5 |
| w2v-bert Pretrain ($\mathcal{L}_u$)+pure Finetune ($\mathcal{L}_s$) | N | 6.6 | 4.3 | 9.9 | 5.0 | 3.8 | 9.1 | 14.6 | 8.1 | 7.7 | 7.8 |
| w2v-bert Pretrain ($\mathcal{L}_u$)+JUST Finetune ($\mathcal{L}$) | N | 6.6 | 4.2 | 9.5 | 5.0 | 4.0 | 9.0 | 15.1 | 7.6 | 7.6 | 7.8 |
| w2v2 Joint Training ($\mathcal{L}_s$+$\beta$($\mathcal{L}_c$+$\alpha\mathcal{L}_d$)) | N | 6.7 | 4.6 | 9.9 | 5.7 | 4.1 | 8.9 | 9.3 | 9.8 | 7.4 | 7.5 |
| JUST ($\mathcal{L}$) | N | 6.8 | 4.6 | 9.9 | 5.7 | 3.9 | 9.1 | 8.6 | 9.1 | 7.2 | 7.3 |
| JUST ($\mathcal{L}$)+pure Finetune ($\mathcal{L}_s$) | N | 6.5 | **4.1** | **9.5** | 5.2 | **3.7** | **8.2** | **8.0** | **6.6** | **6.5** | **6.5** |

**Table 1**: WER(%) results on MLS for different methods. JUST-based methods greatly outperforms the compared baselines. XLSR-53 used external unsupervised data for pretraining. B0 and E3 used external supervised data. Our JUST did not use any external data.

| Method | Ext. | **en** | **de** | **nl** | **fr** | **es** | **it** | **pt** | **pl** | Avg | Avg (w/o **en**) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| JUST ($\beta = 0$) | N | 6.9 | 5.5 | 10.3 | 6.0 | 4.1 | 9.3 | 9.4 | 11.3 | 7.8 | 8.0 |
| JUST ($\beta = 0.03$) | N | 7.4 | 5.0 | 10.3 | 6.3 | 4.3 | 9.3 | 9.1 | 8.7 | 7.5 | 7.6 |
| JUST ($\beta = 0.05$) | N | 6.8 | 5.2 | 9.9 | 5.7 | 4.4 | 9.4 | 8.8 | 9.3 | 7.4 | 7.5 |
| JUST ($\beta = 0.07$) | N | 6.8 | 4.6 | 9.9 | 5.7 | 3.9 | 9.1 | 8.6 | 9.1 | 7.2 | 7.3 |
| JUST ($\beta = 0.1$) | N | 6.8 | 5.8 | 10.0 | 5.8 | 4.1 | 10.3 | 8.6 | 9.7 | 7.6 | 7.8 |

**Table 2**: Weight sensitivity study on $\beta$. When $\beta = 0.07$, the unsupervised loss is roughly the same as the supervised loss, which means balancing the unsupervised and supervised losses can be critical in joint training.

## 4. EXPERIMENTS

### 4.1. Dataset

MLS dataset [12] is used as the benchmark in our experiments. It is derived from read audiobooks of LibriVox. There are 8 languages (namely English (**en**), German (**de**), Dutch (**nl**), French (**fr**), Spanish (**es**), Italian (**it**), Portuguese (**pt**) and Polish (**pl**)), with 44.5k hours of English and 6k hours for other languages combined. Some low-resource language like Polish only has 100 hours. Each utterance is 10-20 seconds long.

### 4.2. Training details

**Architecture** The inputs are 80-d filter bank features. Feature encoder has 2 convolutional layers with filter size (3,3), strides (2,2). The two layers have 128 and 32 channels respectively. Contrastive net consists of 8 Conformer blocks, each with hidden dimensionality 1024, 8 attention heads and convolution kernel size 5. MLM net consists of 16 Conformer blocks with the same configuration. Our decoder uses a 2-layer 768-d LSTM-based RNN-T with 3072 hidden units. $\{c_i\}_{i=1}^T$ from contrastive net and $\{m_i\}_{i=1}^T$ from MLM net are used in computing self-supervised losses after layer normalization. Our codebook has size $V$=1024, with each token of length 1024.

**Masking** To mask $\{z_i\}_{i=1}^T$, we randomly sample 6.5% of all time steps and replace each of the selected time steps and its subsequent 10 time steps with random normal vectors (from $\mathcal{N}(0, 0.1)$). Some spans might overlap.

**Hyperparameters** We train JUST with batch size 1024 on 64 TPUs. Adam optimizer is employed with $\beta_1 = 0.9, \beta_2 = 0.98$ for training. Our global learning rate schedule is the same as [7] but with warm-up steps 5000 and peak learning rate 4e-4. The decoder uses a separate schedule rather than the global one, with 1500 warm-up steps and peak learning rate 7e-4. We set $\alpha = 0.1$ following prior works [5, 7] and $\beta = 0.07$ via tuning with grid search.

**Evaluation** We show the WER for each language, as well as the average WER with or without **en** included.

### 4.3. Compared methods

We compare JUST with several baselines. MLS paper [12] provides competitive monolingual baselines without any LM and with a 5-gram LM. Using LM improves the monolingual performance. XLSR [16] pretrains a w2v2 on 53 languages from MLS, CommonVoice and BABEL, and finetunes the model on MLS. XLSR finetuned on the full set of MLS can outperform some low-resource monolingual baselines like **it**, **pt**, **pl**, but not all (Table 1). We also include transfer learning models, B0 and E3, from [13], which used heavy supervision from external *VoiceSearch* (VS) dataset containing 15 languages. Both B0 and E3 are first trained on VS with supervision and then finetuned on MLS. B0 is a smaller model with 370M parameters and E3 is a larger model with 1B parameters. We also include a B0 model trained from scratch for comparison. Besides these existing baselines from literature, we further train a w2v-bert model from scratch on MLS (JUST with $\beta = 0$), and a 2-stage pretrain-finetune w2v-bert model on MLS without any external data.

### 4.4. Results

For JUST, we either train it from scratch on MLS, or jointly finetune it from a pretrained checkpoint on MLS where the pretraining phase would only optimize the unsupervised loss. Note that compared

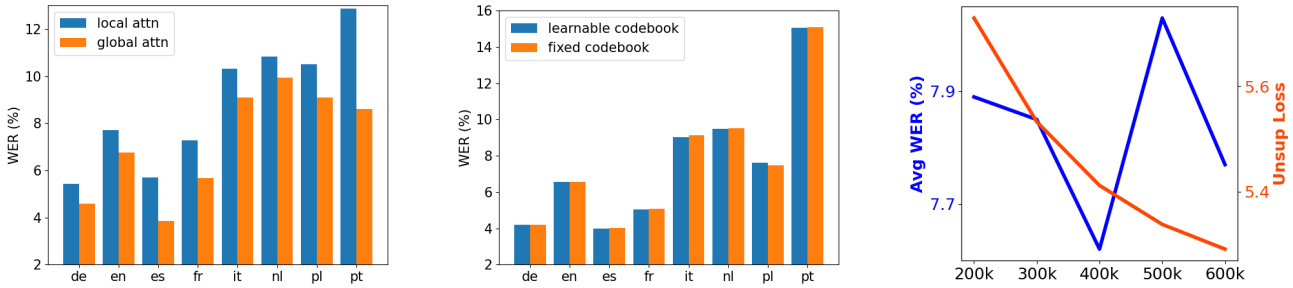


**Fig. 2**: **Left:** Comparison between using local attention with left/right context size 128, and using global attention. Global attention clearly boosts the performance. **Middle:** For JUST finetuning, we either allow the codebook to be updated, or to remain fixed during the whole finetuning. They deliver similar results. **Right:** The average WERs of JUST finetuning ($\beta = 0.01$) from different checkpoints. The checkpoint with smaller unsupervised loss may not lead to the best finetuning results.

to XLSR, our pretraining would incorporate more self-supervised losses. Our JUST has 600M parameters, which is roughly the same scale as B0, XLSR but much smaller than E3.

**Average WER** On the average WER of all 8 languages, all JUST-based methods outperform previous works. In particular, JUST (with $\beta = 0.07$) outperforms the monolingual baseline with 5-gram LM by 33.3%, XLSR-53 by 32.0%, B0 by 18.2%, E3 by 8.8%. Note that E3 is a transfer learning method with much larger size and heavy external supervision. JUST's improvement over E3 validates the effectiveness of our architecture and joint training scheme. If we exclude English WER and compare other languages as in XLSR [16], JUST outperforms monolingual, XLSR-53, B0, E3 by 36.5%, 31.1%, 19.8%, 11.0% respectively. Compared to JUST with $\beta = 0$, JUST with joint training improves the average WER (w/o **en**) by 7.7% (8.8%). To show the necessity of MLM loss in joint training, we further include the results of w2v2 joint training with the objective $\mathcal{L}_s$+$\beta(\mathcal{L}_c$+$\alpha\mathcal{L}_d)$. JUST still performs better on the average WERs and low-resource **es**, **pl**, **pt**.

**Low-resource languages** JUST improves WERs for low-resource languages such as **pl**, **pt**. On **pl**, JUST's WER is less than half of the monolingual WER baseline and roughly half of XLSR's WER. On **pt**, JUST's WER is at least 40% lower than any of XLSR, B0 or E3, which is significant.

**JUST finetune** Two finetuning schemes are attempted. **First**, we take a pretrained checkpoint trained with $\mathcal{L}_u$, and finetune it with JUST objective $\mathcal{L}$. Compared to w2v2 Pretrain+pure Finetune (no MLM loss), it improves on all languages except **pt**. Compared to w2v-bert Pretrain+pure Finetune (with MLM loss), it also improves on **de**, **en**, **fr**, **it**, **nl**, **pl**. It is interesting to compare JUST and JUST Finetune on **pt**, **pl**. Different training schemes lead to different quantized tokens and cause the discrepancy. Empirically, JUST from scratch can better facilitate the low-resource languages and reduce WER of each language to below 10. For JUST finetuning, we set $\beta = 0.01$ to de-weight the unsupervised loss instead of matching with the supervised loss. **Second**, we take a checkpoint from JUST trained from scratch, and finetune it with only supervised loss $\mathcal{L}_s$. It achieves the best average WER, further improving the average WER (w/o **en**) of JUST by 10% (11%). On **de**, **es**, **it**, **nl**, **pl**, **pt**, this scheme outperforms all compared methods and remains competitive on other languages.

**$\beta$ sensitivity** Table 2 also includes the sensitivity study on $\beta$. When $\beta = 0.07$, the unsupervised and the supervised losses are balanced, resulting in the best performance.

**Attention** We compare two attention mechanisms for JUST from scratch: a local attention mechanism with both left and right context 128, and a global attention mechanism with full context. The results are shown in Fig. 2. Global attention clearly outperforms local attention on all languages.

**Codebook** Original w2v2 doesn't update codebook in the finetuning phase. JUST finetuning, however, keeps the unsupervised loss and could further update the codebook. We compare the performance with learnable or fixed codebook during JUST finetuning ($\beta = 0.01$), and find their results are close (Fig. 2). This implies that fixing codebook in JUST finetuning would not degrade the performance. In practice, we also find larger $\beta$ would bias the updates to the codebook, leading to worse results.

**Pretrained checkpoints** Different checkpoints can lead to different downstream performance. The later checkpoints do not necessarily lead to better downstream WERs. To verify this, we finetune multiple pretrained checkpoints and evaluate their finetuning quality. The rightmost subfigure from Fig. 2 shows the constantly descending unsupervised loss $\mathcal{L}_u$, while the downstream average WERs don't follow the same trend.

## 5. CONCLUSION

This work proposes a novel uniform multilingual ASR system for the end-to-end speech recognition on multiple languages. Our method, JUST, is composed of a contrastive module for learning discrete speech representations and an MLM module that performs a masked language modeling task. JUST jointly optimizes the unsupervised contrastive loss and MLM loss, together with the supervised RNN-T loss. Compared to the prevalent 2-stage pretrain-finetune models, JUST-based methods can guide the whole training process with the unsupervised and supervised losses jointly. JUST's performance is validated on a public multilingual ASR dataset, MLS, and outperforms the monolingual baselines, a SOTA 2-stage pretrain-finetune model XLSR, and the latest transfer learning methods, proving the effectiveness of joint training. On low-resource languages, our JUST and its variants can consistently bring gains and boost performance. In the future, we will investigate how the objective function affects the codebook learning, and also explore the joint training with more languages and other unsupervised losses, as well as the tradeoff between unsupervised and supervised components.

## 6. ACKNOWLEDGMENTS

We would like to thank Yu-An Chung, Yotaro Kubo and Weiran Wang for constructive suggestions.

## 7. REFERENCES


[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[3] Zhehuai Chen, Yu Zhang, Andrew Rosenberg, Bhuvana Ramabhadran, Gary Wang, and Pedro Moreno, "Injecting text in self-supervised speech pretraining," *arXiv preprint arXiv:2108.12226*, 2021.

[4] Weiran Wang, Qingming Tang, and Karen Livescu, "Unsupervised pre-training of bidirectional speech encoders via masked reconstruction," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6889–6893.

[5] Yu Zhang, James Qin, Daniel S Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V Le, and Yonghui Wu, "Pushing the limits of semi-supervised learning for automatic speech recognition," *arXiv preprint arXiv:2010.10504*, 2020.

[6] Junwen Bai, Weiran Wang, Yingbo Zhou, and Caiming Xiong, "Representation learning for sequence data with deep autoencoding predictive components," in *International Conference on Learning Representations*, 2020.

[7] Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu, "W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," *arXiv preprint arXiv:2108.06209*, 2021.

[8] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *arXiv preprint arXiv:2106.07447*, 2021.

[9] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[10] Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu, "Recall and learn: Fine-tuning deep pretrained language models with less forgetting," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 7870–7881.

[11] Samuel Kessler, Bethan Thomas, and Salah Karout, "Continual-wav2vec2: an application of continual learning for self-supervised automatic speech recognition," *arXiv preprint arXiv:2107.13530*, 2021.

[12] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert, "Mls: A large-scale multilingual dataset for speech research," in *INTERSPEECH*, 2020.

[13] Bo Li, Ruoming Pang, Tara N Sainath, Anmol Gulati, Yu Zhang, James Qin, Parisa Haghani, W Ronny Huang, Min Ma, and Junwen Bai, "Scaling end-to-end models for large-scale multilingual ASR," *arXiv preprint arXiv:2104.14830*, 2021.

[14] Alexis Conneau and Guillaume Lample, "Cross-lingual language model pretraining," *Advances in Neural Information Processing Systems*, vol. 32, pp. 7059–7069, 2019.

[15] Morgane Riviere, Armand Joulin, Pierre-Emmanuel Mazaré, and Emmanuel Dupoux, "Unsupervised pretraining transfers well across languages," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7414–7418.

[16] Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.

[17] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[18] Yong Cheng, Wei Wang, Lu Jiang, and Wolfgang Macherey, "Self-supervised and supervised joint training for resource-rich machine translation," *arXiv preprint arXiv:2106.04060*, 2021.

[19] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith, "Don't stop pretraining: adapt language models to domains and tasks," *arXiv preprint arXiv:2004.10964*, 2020.

[20] Santiago Pascual, Mirco Ravanelli, Joan Serra, Antonio Bonafonte, and Yoshua Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," *arXiv preprint arXiv:1904.03416*, 2019.

[21] Chaitanya Talnikar, Tatiana Likhomanenko, Ronan Collobert, and Gabriel Synnaeve, "Joint masked CPC and CTC training for ASR," *arXiv preprint arXiv:2011.00093*, 2020.

[22] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[23] Chengyi Wang, Yu Wu, Yao Qian, Kenichi Kumatani, Shujie Liu, Furu Wei, Michael Zeng, and Xuedong Huang, "Unispeech: Unified speech representation learning with labeled and unlabeled data," in *2021 International Conference on Machine Learning*, July 2021.

[24] Srinivasa Raghavan and Kumar Shubham, "Hybrid unsupervised and supervised multitask learning for speech recognition in low resource languages," in *Proc. Workshop on Machine Learning in Speech and Language Processing*, 2021.

[25] Dongseong Hwang, Ananya Misra, Zhouyuan Huo, Nikhil Siddhartha, Shefali Garg, David Qiu, Khe Chai Sim, Trevor Strohman, Françoise Beaufays, and Yanzhang He, "Large-scale asr domain adaptation using self- and semi-supervised learning," *arXiv preprint arXiv:2110.00165*, 2021.

[26] Eric Jang, Shixiang Gu, and Ben Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[27] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al., "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[28] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.