

OPEN IMPLEMENTATION AND STUDY OF BEST-RQ FOR SPEECH PROCESSING

Ryan Whetten¹, Titouan Parcollet², Marco Dinarelli³, Yannick Estève¹

¹ Laboratoire Informatique d’Avignon, Avignon Université, France

² Samsung AI Center, Cambridge, United Kingdom

³ Univervisté Grenoble Alpes, Inria, CNRS, Grenoble INP, LIG, 38000, Grenoble, France

ABSTRACT

Self-Supervised Learning (SSL) has proven to be useful in various speech tasks. However, these methods are generally very demanding in terms of data, memory, and computational resources. BERT-based Speech pre-Training with Random-projection Quantizer (BEST-RQ), is an SSL method that has shown great performance on Automatic Speech Recognition (ASR) while being simpler than other SSL methods, such as wav2vec 2.0. Despite BEST-RQ’s great performance, details are lacking in the original paper, such as the amount of GPU/TPU hours used in pre-training, and there is no official easy-to-use open-source implementation. Furthermore, BEST-RQ has not been evaluated on other downstream tasks aside from ASR and speech translation. In this work, we describe a re-implementation of a Random-projection quantizer and perform a preliminary study with a comparison to wav2vec 2.0 on four downstream tasks. We discuss the details and differences of our implementation. We show that a random projection quantizer can achieve similar downstream performance as wav2vec 2.0 while decreasing training time by over a factor of two.

Index Terms— Self-supervised learning, speech recognition, speaker recognition, keyword spotting.

1. INTRODUCTION

Self-supervised learning (SSL) is a training regime in which a model extracts targets from the input data itself and is trained to predict these targets. SSL systems can thus take advantage of large amounts of unlabeled data in a pre-training stage and use a minimal amount of labeled data to achieve very impressive results on a wide variety of tasks [1]. In the speech domain, SSL has led to state-of-the-art (SOTA) results in tasks such as Automatic Speech Recognition (ASR), automatic emotion recognition (ER), automatic speaker verification (ASV), Spoken Language Understanding (SLU) [1, 2].

However, SSL pre-training is very data, memory, and computationally expensive. For example, authors of wav2vec

2.0, reported using around 2,400 V100 GPU hours and a batch size of 1.6 hours just for the base model [3]. On the high end for very large datasets such as *LeBenchmark*, authors have reported using 54,600 A100 GPU hours for their extra-large model [4].

Despite efforts to enhance the efficiency of other widely used SSL models for speech, such as HuBERT and data2vec [5, 6], the process remains resource intensive. For instance for HuBERT and *data2vec* it requires around 1000 and 700 A100 GPU hours respectively.

One reason of this high cost is due to the Acoustic Feature Extractors (AFE), which is commonly implemented as a series of Convolutional Neural Network (CNN) layers in SOTA architectures, but recent studies have shown that these can be replaced with more efficient alternatives while maintaining performance [7].

A recent model, BERT-based Speech pre-Training with Random-projection Quantizer (BEST-RQ), reduces this cost simply by using Mel filterbanks in place of the AFE, still obtaining impressive results on ASR [8]. Due to this fact, and a few other simplifications (see Section 2.1), BEST-RQ appears to be one of the most efficient SSL methods proposed until now. Currently, no official implementation of BEST-RQ exists, inhibiting the community from having access to a very efficient SSL training method. Furthermore, BEST-RQ performance has only been investigated on two tasks, ASR and speech translation [9]. We aim to fill in these gaps.

In this work, we present our open-source implementation of a random projection quantizer using SpeechBrain [10] and conduct an initial examination by comparing BestRQ to wav2vec 2.0.¹ We analyze the pre-training time as well the performance on the following downstream tasks: automatic speech recognition (ASR), automatic speaker verification (ASV), intent classification (IC), and emotion recognition (ER). Results show that a random projection quantizer can achieve similar performance on these various downstream tasks as wav2vec 2.0 with the added benefit of reducing SSL pre-training time by more than half. We believe our open-source implementation can serve as a starting point for

This work received funding from the French ANR E-SSL project (grant N°ANR-22-CE23-0013) and was performed using HPC resources from GENCI-IDRIS (projects A0131013821 and AD011014732)

¹The code for our BEST-RQ implementation is available at <https://github.com/speechbrain/speechbrain/tree/develop/recipes/LibriSpeech>

furthering research, facilitating exploration into various SSL architectures for speech.

2. BACKGROUND

While there are other effective SSL models for speech processing, such as HuBERT and data2vec [5, 6], wav2vec 2.0 seems more wide-spread and used by the speech community, with several freely available tools and models, making thus a comparison straightforward. In this section, we give a more detailed comparison of wav2vec 2.0 and BEST-RQ.

2.1. wav2vec 2.0 vs. BEST-RQ

BEST-RQ and wav2vec 2.0 have a few major differences, one of these being that **BEST-RQ does not operate on the raw audio**. Instead, BEST-RQ operates on 80-dimensional log Mel filterbank coefficients with two CNN layers for the Acoustic Feature Extractor (AFE). In contrast, wav2vec 2.0 operates on the raw audio using seven CNN layers for the AFE.

Using a hand-crafted AFE such as Mel filterbanks significantly reduces the amount of memory and computations needed to train the model. However, this may limit the performance of the model as filterbanks are compressed views of speech based on psycho-acoustics (i.e. based on human expertise) and, consequentially, could remove acoustic cues that a non-hand-crafted AFE can learn.

A second major difference is that **BEST-RQ uses a randomly initialized linear layer and one codebook for quantizing and discretizing the audio**. These components are both fixed throughout training. The Mel filterbanks are projected with the linear layer, then the index of the nearest codebook entry to the projection is used as the target. The nearest codebook entry is found by taking the argmin of the normalized distance between each entry in the codebook and the projection.

Afterward a mask is applied to a portion of Mel filterbanks and the objective of the model is to guess the correct targets for the masked sections. This is treated as a classification task using cross entropy to calculate the loss.

Conversely, in wav2vec 2.0, there are two learned codebooks which are trained via a gumbel softmax. The loss function for wav2vec 2.0 is more complicated consisting of a combination of a contrastive loss (i.e. identifying the correct label among a portion of randomly selected false labels) and a diversity loss (to prevent the model from collapsing to using just one entry in the codebook).

One last major difference is that **BEST-RQ uses conformer layers** instead of transformer layers. Note that although conformer layers with wav2vec 2.0 has been studied [11], for this work we only study the wav2vec 2.0 standard architecture as our goal is not to prove one methodology is better than another, it is instead to show that our implementation works as expected and compares favorably to a well-

known alternative. See Figure 1 for a simple diagram comparing these two architectures.

[8] showed that good ASR performance (in terms of word error rates) can be achieved without having to train an AFE or codebooks. Theoretically, this significantly reduces the complexity of the model, the loss function and the backward pass. As the pre-training time was not reported in the original BEST-RQ paper, we aim to show how this affects pre-training empirically and offer evaluations on ASR and three other downstream tasks. For reproducibility and to allow others to study effects of using a random projection quantizer, we make our implementation open source.

3. EXPERIMENTS

For our experiments we pre-train a wav2vec 2.0 model and our BEST-RQ implementation and then evaluate on a subset of the MP3S benchmark tasks [12]. In the next sections we describe the pre-training environment, the architecture of the models, and the downstream tasks.

3.1. Pre-training Setting

For pre-training, we use 960 hours of LibriSpeech (i.e. the *train-clean-100*, *train-clean-360* and *train-other-500* splits) [13]. We train all models for 42 epochs or roughly 200k steps using eight Tesla V100 GPUs. We save a checkpoint at epoch 21, around 100k steps, to evaluate performance throughout training. We denote the difference between these versions of the model by naming them *100k* and *200k*.

We use the same dynamic batching settings for both wav2vec 2.0 and BEST-RQ with the max batch size being 100 seconds. Because we use eight GPUs, the total batch size is 800 sec or about 13.33 minutes. We chose this batch size to be able to run our experiments on small GPUs and within a limited amount of time using only a few hundred GPU hours.

3.2. Model Settings

We use the wav2vec 2.0 base model architecture [3] as a baseline, using the implementation of SpeechBrain. We only modified the dynamic batch settings to have a max batch size of 100 seconds.

For our BEST-RQ implementation, we follow overall the author’s description, but we make a **few key changes** found through some preliminary experiments, and that revealed very important for performance with the relatively small batch size (13 min. vs. 18hours), small amount of pre-training data (960 hours vs. for example 12 million [9]), and number of GPUs. **We shrink the number of conformer layers from 24 to 12.** This was done in order to match the same number of layers as wav2vec 2.0 base model, and allowed the model to fit on lower-end GPUs. **We added layer dropout of 0.05, we reduced the learning rate to 0.0008 and lastly we increase the mask ratio to about 60% of the audio** (i.e. 15% of frames

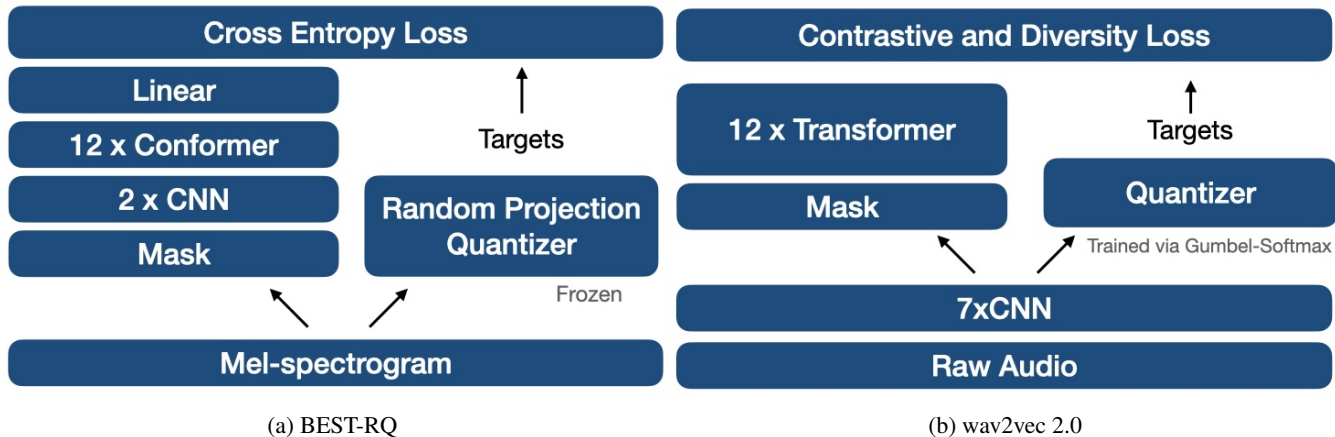


Fig. 1: Diagrams of our BEST-RQ and wav2vec 2.0 architecture. BEST-RQ operates on mel-spectrograms, uses a static quantizer and conformer layers. On the other hand, wav2vec 2.0 operates on raw audio, trains the quantizer, and uses transformer layers.

are randomly selected to be masked together with the following three frames). We describe some results from our initial experiments in Section 4.2.

3.3. Downstream Tasks

For the downstream tasks we follow the methodology in the MP3S [12] and the SUPERB [2] benchmarks, that is the SSL model is frozen and the input to the downstream task specific model is a weighted sum of the hidden states from the pre-trained model. In the MP3S, each downstream task has two probes, of which we select one per task due to time and resource constraints.

For **Automatic Speech Recognition (ASR)** we use the *train-clean-100/dev-clean* for training and validation respectively and then evaluate on the *test-clean* and *test-other splits* from LibriSpeech [13].

For this task, we use a 2-layered BiLSTM followed by a linear layer, softmax layer, and then CTC loss [12]. The BiLSTM has a size of 1024 and a dropout of 0.2. The metric we use to measure performance is the Word Error Rate (WER). We report the WER with and without applying the official 4-gram ARPA LM [12].²

For **Automatic Speaker Verification (ASV)** we use Voxceleb1 [14]. This dataset is made up of utterances of over 1,000 celebrities gathered from youtube. The task is a binary classification task where given two audio files, the model has to determine whether the speaker is the same or not.

We use the ECAPA-TDNN probe from the MP3S benchmark. An ECAPA-TDNN uses a time delayed neural network, statistical pooling with temporal and channel self-attention in order for generating fixed length high quality speaker embeddings [15]. To measure the performance of ASV we use the equal error rate (EER).

²Available at openslr.org/11/

For **Intent Classification (IC)** we use SLURP [16], which is known to be more challenging than other datasets, consisting of 177 speakers from 72k audio files adding up to 58 hours of audio. The task is to classify a given utterance into 1 of 18 tasks or scenarios such as *email*, *calendar*, or *play* (as in *play next song*).

For the probe, again we use one from MP3S. This probe is a BiLSTM with a size of 1024 that is followed by a linear layer, a statistic pooling layer, and then a linear layer for the final classification. The evaluation metric for this task is the accuracy.

For **Emotion Recognition (ER)** we use the IEMOCAP dataset [17]. This dataset contains about 12 hours of data from 10 speaker actors performing scripts with four different emotions (neutral, happy, sad and angry). The performance is measured with a 10-fold cross validation. Similar to the ASV task we use the ECAPA-TDNN probe from the MP3S benchmark.

In order to explore the capabilities of BEST-RQ not as just a model for speech embedding but as a model that can be fine-tuned (i.e. not frozen) for a downstream task, we perform one more experiment where we fine-tune the model using a vanilla feed-forward neural network with CTC loss. We fine-tune using the *train-clean-100* and then evaluate on the *test-clean* and *test-other splits* from LibriSpeech with and without the 4-gram ARPA language model.

4. RESULTS

In this section we report results on the pre-training time, downstream tasks, and on some initial experiments varying the masking ratio and the size of the codebook in our BEST-RQ implementation.

Model / Task Metric	# Par.	GPU hours	LibriSpeech train-100 ASR				VoxCeleb1	SLURP	IEMOCAP
			WER ↓				EER ↓	Acc. ↑	Acc. ↑
			Clean	Clean LM	Other	Other LM	ASV	IC	ER
W2V2 100k	90.9M	130	16.26	10.63	40.17	30.83	4.56	72.8	60.9
W2V2 200k	90.9M	262	13.89	9.45	33.55	25.49	3.83	74.5	63.0
BRQ 100k	83.0M	54	16.79	10.79	38.09	28.31	3.84	74.3	61.3
BRQ 200k	83.0M	109	15.11	9.76	34.06	24.74	3.53	74.8	63.8

Table 1: Results on downstream tasks at 100k and 200k steps. BEST-RQ performs similarly to wav2vec but with less than half of the training time. Training time and number of parameters are shown under *GPU hours* and *# Par.* respectively.

4.1. MP3S Downstream Tasks and Fine-Tuning

Among downstream experiments, reported in Table 1, wav2vec 2.0 proved to perform better than BEST-RQ without a language model on the MP3S LibriSpeech *train-100* ASR task. However, with the language model, BEST-RQ performs very close to wav2vec 2.0 on the *test-clean* and even slightly better on the *test-other* split. Additionally, when we fine-tune the model for ASR, BEST-RQ performs better than wav2vec 2.0. On the other tasks, ASV, IC and ER, BEST-RQ performs slightly better than wav2vec 2.0, while pre-training approximately 2.4 times faster than wav2vec 2.0.

Examining the difference between the models at 100k and 200k training steps, in Table 2, we note that in every case wav2vec 2.0 has a greater improvement in performance. For example on IC, the wav2vec 2.0 200k model improves by 1.7% over the 100k model, while BEST-RQ improves by 0.5%. We believe this is due to the fact that the AFE in wav2vec 2.0 has to be trained, which results in slower convergence in terms of both time and number of steps.

Model	Fine-Tune LibriSpeech train-100			
	Clean	Clean LM	Other	Other LM
W2V2 100k	16.42	10.29	36.62	27.25
W2V2 200k	13.47	8.73	29.64	21.92
BRQ 100k	14.59	9.00	31.49	23.04
BRQ 200k	12.21	7.78	26.81	19.67

Table 2: Results of fine-tuning on ASR with LibriSpeech *train-100*.

4.2. Impact of the Masking Ratio

We train all models for 18 epochs or about 87k steps on 4 x 11GB 2080Ti GPUs and we only looked at the ASR downstream performance on the validation set (i.e. the *dev-clean* split of LibriSpeech). We combined the variation of the masking ratio with the variation of the codebook size when using 1% and 10% masking ratios.

As shown in Table 3, shrinking the codebook (CB) from the original 8192 to 1024 does not seem to change the downstream performance in any consistent or significant way. At a masking ratio of 1% the smaller codebook performs slightly better, while at 10% the smaller codebook performs slightly

worse. Instead, increasing the amount of masked frames has a great impact, decreasing the WER on the *dev-clean* split by about 15% (from a WER around 35% down to a bit above 20%) as the amount of frames selected for masking increased from 1% to 12%.

Mask %	Dev-Clean WER	CB
1%	34.08	1024
1%	36.45	8192
5%	25.24	8192
10%	21.10	1024
10%	20.68	8192
12%	20.11	8192

Table 3: Impact of changing the masking ratio and the codebook size. The mask % concerns the start frame index chosen for the mask. Since the following three frames are also masked the actual masking ratio is 4 times as big.

5. DISCUSSION AND CONCLUSION

In this work we described our open-source implementation of a random projection quantizer and we compared it to wav2vec 2.0 in terms of pre-training time and performance on various downstream tasks. BEST-RQ demonstrates comparable performance to wav2vec 2.0 while speeding up pre-training time by more than half. We will release our code in the *Speech-Brain* toolkit [10], and believe our open-source implementation can allow researchers to further investigate these methodologies and explore diverse facets of SSL models for speech.

We hypothesize that because BEST-RQ starts with mel-spectrograms and is about 8M parameters smaller, BEST-RQ converges much faster than wav2vec 2.0 resulting in comparable performance in 200k. The differences in performance between the 100k and 200k models suggest that wav2vec 2.0 model could surpass BEST-RQ given more time or other training settings (such as a larger batch size).

Nevertheless, we believe our results to be insightful on the capabilities of these methods given only a few hundred GPU hours and leave more in depth experiments for future work.

6. ACKNOWLEDGMENT

We would like to thank the authors of BEST-RQ [8] for their helpful discussion.

7. REFERENCES

- [1] Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, et al., “Self-supervised speech representation learning: A review,” *IEEE Journal of Selected Topics in Signal Processing*, 2022.
- [2] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al., “Superb: Speech processing universal performance benchmark,” *arXiv preprint arXiv:2105.01051*, 2021.
- [3] 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, pp. 12449–12460, 2020.
- [4] Titouan Parcollet, Ha Nguyen, Solene Evain, Marcely Zanon Boito, Adrien Pupier, Salima Mdhaffar, Hang Le, Sina Alisamir, Natalia Tomashenko, Marco Dinarelli, et al., “Lebenchmark 2.0: a standardized, replicable and enhanced framework for self-supervised representations of french speech,” *arXiv preprint arXiv:2309.05472*, 2023.
- [5] William Chen, Xuankai Chang, Yifan Peng, Zhaoheng Ni, Soumi Maiti, and Shinji Watanabe, “Reducing barriers to self-supervised learning: Hubert pre-training with academic compute,” *arXiv preprint arXiv:2306.06672*, 2023.
- [6] Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli, “Efficient self-supervised learning with contextualized target representations for vision, speech and language,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 1416–1429.
- [7] Titouan Parcollet, Shucong Zhang, Rogier van Dalen, Alberto Gil CP Ramos, and Sourav Bhattacharya, “On the (in) efficiency of acoustic feature extractors for self-supervised speech representation learning,” in *Inter-speech 2023*, 2023.
- [8] Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu, “Self-supervised learning with random-projection quantizer for speech recognition,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 3915–3924.
- [9] Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, et al., “Google usm: Scaling automatic speech recognition beyond 100 languages,” *arXiv preprint arXiv:2303.01037*, 2023.
- [10] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al., “Speechbrain: A general-purpose speech toolkit,” *arXiv preprint arXiv:2106.04624*, 2021.
- [11] 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.
- [12] Salah Zaiem, Youcef Kemiche, Titouan Parcollet, Slim Essid, and Mirco Ravanelli, “Speech self-supervised representation benchmarking: Are we doing it right?,” *arXiv preprint arXiv:2306.00452*, 2023.
- [13] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.
- [14] Arsha Nagrani, Joon Son Chung, and Andrew Senior, “Voxceleb: a large-scale speaker identification dataset,” *arXiv preprint arXiv:1706.08612*, 2017.
- [15] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck, “Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification,” *arXiv preprint arXiv:2005.07143*, 2020.
- [16] Emanuele Bastianelli, Andrea Vanzo, Pawel Swietojanski, and Verena Rieser, “SLURP: A spoken language understanding resource package,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, Eds., Online, Nov. 2020, pp. 7252–7262, Association for Computational Linguistics.
- [17] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan, “Iemocap: Interactive emotional dyadic motion capture database,” *Language resources and evaluation*, vol. 42, pp. 335–359, 2008.