Towards efficient self-supervised representation learning in speech processing

Anonymous ACL submission

Abstract

Self-supervised learning has achieved impressive results in speech processing, but current models are computationally expensive, generating environmental concerns because of their high energy consumption. Therefore, we propose an efficient self-supervised approach to address high computational costs, using a single GPU during 24 to 48 hours of pretraining. The proposed approach combines linear, convolutional, and self-attention layers with several optimizations, including dynamic batching, flash attention, mixed-precision training, gradient accumulation, and acoustic feature extraction with input preprocessing. Computational cost estimations for our proposed model represent up to two orders of magnitude improvements in computational efficiency against existing speech models.

1 Introduction

Self-supervised models generate impressive results when learning latent representations, but their training is computationally expensive (Peng et al., 2023). Yet, their results in speech processing are astounding because downstream tasks strongly benefit from their learned representations (Mohamed et al., 2022; Parcollet et al., 2023b).

Self-supervised approaches for speech representation learning can be based on consistency or self-training (Zhang et al., 2020). Whether using consistency or self-training, large training costs represent a challenge. Indeed, most existing models require several GPUs for days to pretrain their neural architectures. This requirement causes several limitations. First, it hinders the training and deployment of speech models in computing platforms with low resources, such as edge devices and mobile platforms (Gaol et al., 2023; Mohamed et al., 2022). Secondly, reproducibility is challenging, as few labs can afford large computational resources (Lin et al., 2022). Last but not least, it creates environmental issues because of the high energy consumption during training (Parcollet et al., 2023b).

To address those limitations, we propose an efficient self-supervised model to learn speech representations. Instead of focusing on the model performance in downstream tasks, the proposed model focuses primarily on computational costs, limiting the resources available for pretraining. We set a pretraining limit based on cramming (Geiping and Goldstein, 2023): we use a single GPU for 24 to 48 hours to train the model.

2 Related work

Several models have been recently proposed for self-supervised learning of speech representations, including CombinedSSL (Zhang et al., 2020), Mockingjay (Liu et al., 2020), Spiral (Huang et al., 2022), Data2vec2(Baevski et al., 2023), and DinoSR (Liu et al., 2023a). But two approaches have clearly emerged (Mohamed et al., 2022): Hidden unit BERT (HuBERT) (Hsu et al., 2021) and wav2vec2 (Baevski et al., 2020b). However, self-supervised models are quite costly, requiring a lot of computational resources for training. One alternative to reduce training costs is knowledge distillation (Allen-Zhu and Li, 2020), where a small student model learns from a large teacher model, which has been pretrained previously (Peng et al., 2023).

Using knowledge distillation, LightHuBERT (Wang et al., 2022) improves HuBERT with a once-for-all transformer model. The teacher is a HuBERT base model, while the student learns by predicting masked inputs in an iterative process. The transformer in LightHuBERT comprises subnets with sharable weights and several configuration parameters, enabling a random search to adjust the model to different resource constraints.

The student architecture in knowledge distillation methods is manually designed, and it does not change during training. However, modifying student architectures can have a considerable impact on model results, even for student architectures with similar sizes (Ashihara et al., 2022). Therefore, a joined distillation and pruning approach for speech SSL has been recently proposed, using HuBERT (DPHuBERT) or WavLM (DPWavLM) as the teacher models (Peng et al., 2023).

Yet, knowledge distillation approaches need a pretrained teacher model because student models can not be trained standalone (Chen et al., 2023). Thus, computational costs do not improve as they should include teacher model training. In contrast, MelHuBERT (Lin et al., 2022) proposes a simplified version of HuBERT that has twelve self-attention layers and a weighted sum of all the layers for downstream tasks. The input is a 40-dimensional Mel log spectrogram, so input sequences are shorter, reducing the multiplication and addition calculations by 33% (Lin et al., 2022).

There are also efforts to improve the wav2vec architecture. Proposed approaches improving wav2vec include squeezed and efficient wav2vec2 with disentangled attention (SEW-D) (Wu et al., 2022) and stochastic squeezed and efficient wav2vec2 (S-SEW) (Vyas et al., 2022).

Despite existing efforts to improve self-supervised model efficiency, there is still room to reduce the computational costs of self-supervised models. Computational costs create challenges when using these models in mobile devices and for training on very large datasets (Mohamed et al., 2022; Parcollet et al., 2023b). They also hinder the development of new approaches, the study of other training recipes, and the reproduction of experimental results, as few researchers can afford the cost (Chen et al., 2023; Lin et al., 2022; Parcollet et al., 2023b; Wang et al., 2023). Besides, computational costs have environmental implications, as training requires considerable amounts of energy (Parcollet et al., 2023b).

Likewise, few existing self-supervised models use half-precision numbers, even though this technique can half the memory requirements and accelerate the arithmetic computations on recent GPUs (Micikevicius et al., 2018). A similar issue happens with dynamic batching (Gaol et al., 2023; Tyagi and Sharma, 2020), a procedure that avoids wasting computing resources on the padded portion of speech mini-batches. Also, most models use standard self-attention layers, though efficient alternatives have been recently proposed, without using approximations (Dao et al., 2022; Parcollet et al., 2023a).

3 Efficient self-supervised approach

In this section, we describe our proposed efficient self-supervised learning (ESSL) model and the optimizations used to improve model efficiency.

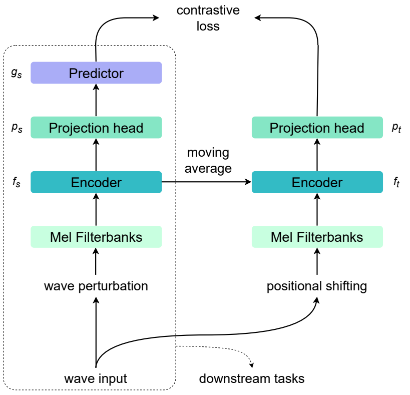

Figure 1: Neural architecture for our proposed ESSL approach, based on a teacher – student configuration (Huang et al., 2022).

3.1 Model architecture

The architecture uses a teacher – student configuration based on recent work for speech processing (Huang et al., 2022). The student part comprises an encoder, a projection head, and a predictor, while the teacher part comprises an encoder and a projection head (Figure 1). Following a conformer configuration (Gulati et al., 2020), the encoder has 3 convolutional layers, followed by 2 self-attention layers, 2 convolutional layers, and 10 self-attention layers. Projection heads are linear layers, and the predictor has 3 convolutional layers (Huang et al., 2022). Self-attention layers use relative position embeddings to better capture the sequence ordering of input sequences (Chen et al., 2022).

Pretraining relies on a contrastive loss to force the student latent representation to converge to the latent representation of the teacher part of the model, updating teacher weights with an exponential moving average of student weights (Chen et al., 2020; Huang et al., 2022).

Regularization for the proposed model includes dropout, SpecAugment (Park et al., 2019), random positional shifting (Huang et al., 2022), and multicondition training (Chiba et al., 2019) through noise addition. For noise addition, audio data comes from the DNS 2021 challenge (Reddy et al., 2021), adding noise audio to the utterances in the input dataset. Noise addition is performed randomly, with a probability of 0.5 (Huang et al., 2022).

3.2 Model optimizations

Optimizations in our proposed model include flash attention (Dao et al., 2022), mixed precision training (Micikevicius et al., 2018), dynamic batching (Tyagi and Sharma, 2020), gradient accumulation (Huang et al., 2023), and acoustic feature extraction (AFE) with input preprocessing (Parcollet et al., 2023b). AFE comprises the first part of the neural model, processing the input signal before feeding it to the subsequent layers. The best-performing approaches for AFE combine Mel Filterbanks for preprocessing the raw waveform before the convolutional module (Parcollet et al., 2023b), as we do in ESSL.

Batch sizes have a considerable impact on training performance (Chen et al., 2023; Hsu et al., 2021). To deal with the high memory requirements of large batch sizes with a single GPU, gradients are accumulated for a few training steps before applying them to update the parameters of the model (Huang et al., 2023). This approach enables the increase in batch size to get close to batch sizes used in large models (Liu et al., 2023a).

Another optimization involving training batches is dynamic batching (Ravanelli et al., 2021). Based on the duration of each audio file, dynamic batching packages one or several files into a single batch, keeping the total batch duration under a specified maximum duration. By doing so, dynamic batching minimizes the amount of padding that fixed batch sizes must use. This optimization reduces the amount of RAM required to train a model. It also eliminates the GPU iterations wasted when processing the padding data in fixed batch sizes.

Concerning the number format for model parameters and data, mixed precision training uses the floating point 16 (FP16) format. FP16, also known as half-precision, diminishes the size of the model and the batches, using less RAM during training than the floating point 32 (FP32) commonly used in computations. FP16 also enables faster training in the GPU, without affecting the convergence of the model (Micikevicius et al., 2018; Narayanan et al., 2021).

Lastly, FlashAttention (Dao et al., 2022) improves the efficiency of self-attention layers by focusing on the optimization of the input-output (IO) memory operations in the GPU. In general, GPUs have two kinds of memories. A small SRAM is associated with each kernel, and a large high-bandwidth memory, which is slower and is shared between all the kernels. Memory-intensive operations, like the matrix operation of the self-attention layers, have their bottleneck at the read-write RAM access. In contrast, compute-intensive operations have their bottleneck in the number of arithmetic operations that must be realized. As self-attention is primarily a memory-intensive operation, FlashAttention reduces the number of IO operations by tiling, assigning a matrix operation to a single kernel, and saving some results from the forward pass to share in the subsequent backward pass.

4 Results and discussion

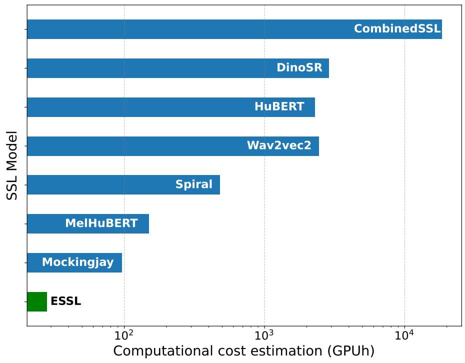

Figure 2: Cost estimation for pretraining speech SSL models. ESSL represents a remarkable reduction in computational costs against existing models.

All experiments run on a single GPU, an NVIDIA GeForce RTX 3090 Ti with 24 GB of memory. Considering training data, LibriSpeech 960h provides speech utterances for unsupervised pretraining. Finetuning for Automatic Speech Recognition (ASR) is performed with LibriSpeech 100h (Panayotov et al., 2015), using a CTC loss (Yan et al., 2023). Regarding training configuration, pretraining requires 60k iterations, which is equivalent to 15k pretraining steps because we do 4 gradient accumulations. The learning rate warms up the first 8% of iterations to a maximum of 3e-4. For finetuning, 160k iterations are performed. This is equivalent to 40k finetuning steps with 4 gradient accumulations. The learning rate warms up the first 10% of iterations to a maximum of 3e-5 (Huang et al., 2022).

Efficiency gains of ESSL are remarkable (Figure 2). Though metrics degrade against large speech models (Table 1), the computational cost estima-

SSL Model	ASR
Mockingjay (Liu et al., 2020)	15.48
wav2vec (Schneider et al., 2019)	11.00
vq-wav2vec (Baevski et al., 2020a)	12.80
wav2vec2 Base (Baevski et al., 2020b)	4.79
HuBERT Base (Hsu et al., 2021)	4.79
Spiral Base (Huang et al., 2022)	3.30
WavLM Base (Chen et al., 2022)	3.40
CombinedSSL (Zhang et al., 2020)	1.60
ESSL	10.69

Table 1: WER for LibriSpeech test-clean dataset (Yang et al., 2021). Models are pretrained with LibriSpeech 960h. ASR results use a language model for decoding.

tion represents a fifth of recent work (Lin et al., 2022), diminishing from 150 GPUh to only 28 GPUh, and about a third of recent work (Liu et al., 2020). When doing a comparison against large models, their computational cost estimations are around one or two orders of magnitude larger. For example, Spiral takes 480 GPUh, which is 15 times larger than our proposed approach. Similarly, CombinedSSL takes 18432 GPUh, which is 576 times larger than ESSL.

As mentioned, batch size is crucial for training speech processing models (Chen et al., 2023). Using dynamic batching, half-precision, and gradient accumulation enables ESSL to get close to the batch sizes used in large speech models – but using one GPU only. The batch size has 18 minutes of audio data. With 4 gradient accumulations, it gets to 72 minutes. This size is close to batch sizes used in recent speech models, such as 47 minutes in HuBERT, 96 minutes in wav2vec2, or 187 minutes in WavLM (Liu et al., 2023a).

Perturbations on input speech sequences are also crucial for the performance of ESSL. Removing them makes WER degrade from 29.91% to 40.08% (Table 2). This drop in performance indicates the importance of SpecAugment, random positional shifting, and multicondition training through noise addition in the pretraining process.

Other experiments to analyze ESSL include random initialization and MelHuBERT configuration. For experiments with MelHuBERT configuration, we used 40 Mel Filterbanks, with a 20ms hop length (Lin et al., 2022). Though training steps can be up to 36% faster given shorter input sequence lengths, WER drops considerably, going from 29.91% down to 51.09%. Concerning random initialization, we discarded pretrained weights and finetuned from a model with random weights. Results suggest finetuning only is not enough for speech processing. A WER of 99.7% highlights the importance of pretraining in final ESSL results.

Configuration	dev-other	dev-clean
ESSL	**28.18**	**10.38**
- w/o perturbations	40.08	17.88
- w/ 40 Mel Filterbanks	51.09	26.41
- random initialization	99.70	99.78

Table 2: Analysis of different configurations for ESSL. Results include WER performance on LibriSpeech dev-other and dev-clean datasets.

5 Limitations

Very-low data settings are challenging. The limited availability of data hinders research in speech processing for under-resourced languages (Liu et al., 2023b; Shi et al., 2021). We tested finetuning ESSL for ASR with the Librilight dataset (Kahn et al., 2020). Librilight has 10 hours, 1 hour, and 10 minutes datasets to finetune models, in contrast with the 100 hours available in LibriSpeech. Results indicate ESSL struggles in very-low data settings, with a WER of 99.97% in LibriSpeech dev-other, a degradation too high to perform ASR for under-resourced languages.

6 Conclusion

In this work, we proposed ESSL, an efficient approach for self-supervised learning of speech representations. ESSL addresses high computational costs by combining several model optimizations and fixing a limit on computational resources available for pretraining. Estimations of computational cost reduction reveal up to two orders of magnitude improvements against existing speech SSL models. Overall, ESSL is a step in the process of reducing computational costs in SSL models, enabling their training in edge devices, facilitating the development of new approaches, and making them more environmentally friendly.

For future work, we will investigate our efficient approach for other speech processing tasks, including intent classification, keyword spotting, query by example, and other downstream tasks. We will also explore architectural modifications to improve model performance in very-low data settings.

References

Zeyuan Allen-Zhu and Yuanzhi Li. 2020. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*.

Takanori Ashihara, Takafumi Moriya, Kohei Matsuura, and Tomohiro Tanaka. 2022. Deep versus wide: An analysis of student architectures for task-agnostic knowledge distillation of self-supervised speech models. In *Interspeech*.

Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli. 2023. Efficient self-supervised learning with contextualized target representations for vision, speech and language. In *ICML*.

Alexei Baevski, Steffen Schneider, and Michael Auli. 2020a. vq-wav2vec: Self-supervised learning of discrete speech representations. In *ICLR*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020b. wav2vec 2.0: A framework for self-supervised learning of speech representations. *NeurIPS*.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, (6).

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*.

William Chen, Xuankai Chang, Yifan Peng, Zhaoheng Ni, Soumi Maiti, and Shinji Watanabe. 2023. Reducing barriers to self-supervised learning: Hubert pre-training with academic compute. In *Interspeech*.

Yuya Chiba, Takashi Nose, and Akinori Ito. 2019. Multi-condition training for noise-robust speech emotion recognition. *Acoustical Science and Technology*, (6).

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *NeurIPS*.

Yan Gaol, Javier Fernandez-Marques, Titouan Parcollet, Pedro PB de Gusmao, and Nicholas D Lane. 2023. Match to win: Analysing sequences lengths for efficient self-supervised learning in speech and audio. In *IEEE SLT Workshop*.

Jonas Geiping and Tom Goldstein. 2023. Cramming: Training a language model on a single gpu in one day. In *ICML*.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech*.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

Wenyong Huang, Zhenhe Zhang, Yu Ting Yeung, Xin Jiang, and Qun Liu. 2022. Spiral: Self-supervised perturbation-invariant representation learning for speech pre-training. In *ICLR*.

Zimeng Huang, Bo Jiang, Tian Guo, and Yunzhuo Liu. 2023. Measuring the impact of gradient accumulation on cloud-based distributed training. In *IEEE/ACM International Symposium CCGrid*.

Jacob Kahn, Morgane Riviere, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. 2020. Librilight: A benchmark for asr with limited or no supervision. In *ICASSP*.

Tzu-Quan Lin, Hung-yi Lee, and Hao Tang. 2022. Melhubert: A simplified hubert on mel spectrogram. *arXiv preprint arXiv:2211.09944*.

Alexander H Liu, Heng-Jui Chang, Michael Auli, Wei-Ning Hsu, and James R Glass. 2023a. Dinosr: Self-distillation and online clustering for self-supervised speech representation learning. *arXiv preprint arXiv:2305.10005*.

Andy T Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee. 2020. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP*.

Zoey Liu, Justin Spence, and Emily Prud'Hommeaux. 2023b. Investigating data partitioning strategies for crosslinguistic low-resource asr evaluation. In *EACL*.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. 2018. Mixed precision training. In *ICLR*.

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. 2022. Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, et al. 2021. Efficient large-scale language model training on gpu clusters using megatron-lm. In *ICHPC-NSA*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *ICASSP*.

Titouan Parcollet, Rogier van Dalen, Shucong Zhang, and Sourav Bhattacharya. 2023a. Sumformer: A linear-complexity alternative to self-attention for speech recognition. *arXiv preprint arXiv:2307.07421*.

Titouan Parcollet, Shucong Zhang, Rogier van Dalen, Alberto Gil CP Ramos, and Sourav Bhattacharya. 2023b. On the (in) efficiency of acoustic feature extractors for self-supervised speech representation learning. In *Interspeech*.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech*.

Yifan Peng, Yui Sudo, Shakeel Muhammad, and Shinji Watanabe. 2023. Dphubert: Joint distillation and pruning of self-supervised speech models. In *Interspeech*.

Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, et al. 2021. Speechbrain: A general-purpose speech toolkit. *arXiv preprint arXiv:2106.04624*.

Chandan KA Reddy, Harishchandra Dubey, Vishak Gopal, Ross Cutler, Sebastian Braun, Hannes Gamper, Robert Aichner, and Sriram Srinivasan. 2021. Icassp 2021 deep noise suppression challenge. In *ICASSP*.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. In *Interspeech*.

Jiatong Shi, Jonathan D Amith, Rey Castillo García, Esteban Guadalupe Sierra, Kevin Duh, and Shinji Watanabe. 2021. Leveraging end-to-end asr for endangered language documentation: An empirical study on yolóxochitl mixtec. In *EACL*.

Sahil Tyagi and Prateek Sharma. 2020. Taming resource heterogeneity in distributed ml training with dynamic batching. In *ICACSOS*.

Apoorv Vyas, Wei-Ning Hsu, Michael Auli, and Alexei Baevski. 2022. On-demand compute reduction with stochastic wav2vec 2.0. *arXiv preprint arXiv:2204.11934*.

Rui Wang, Qibing Bai, Junyi Ao, Long Zhou, Zhixiang Xiong, Zhihua Wei, Yu Zhang, Tom Ko, and Haizhou Li. 2022. Lighthubert: Lightweight and configurable speech representation learning with once-for-all hidden-unit bert. In *Interspeech*.

Sid Wang, John Nguyen, Ke Li, and Carole-Jean Wu. 2023. Read: Recurrent adaptation of large transformers. *arXiv preprint arXiv:2305.15348*.

Felix Wu, Kwangyoun Kim, Jing Pan, Kyu J Han, Kilian Q Weinberger, and Yoav Artzi. 2022. Performance-efficiency trade-offs in unsupervised pre-training for speech recognition. In *ICASSP*.

Brian Yan, Siddharth Dalmia, Yosuke Higuchi, Graham Neubig, Florian Metze, Alan W Black, and Shinji Watanabe. 2023. Ctc alignments improve autoregressive translation. In *EACL*.

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. 2021. Superb: Speech processing universal performance benchmark. In *Interspeech*.

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