General Purpose Audio Embeddings from Encodec and Mobile Nets

Davin Lawrence
davin.lawrence@utexas.edu

Jasper Lee
leejasper@utexas.edu

Alan Baade
abaade@utexas.edu

Abstract

We introduce a model for general purpose audio embeddings based on using EnCodec Units. Our approach takes as input a raw stream of quantized input, unembeds it, and feeds it into a downstream MobileNet. This approach allows for low-bandwidth streamed inputs to be calculated on-device for low energy and storage size. We evaluate our approach on a comprehensive set of datasets and compare to prior work to see where this usecase is best satisfied. We observe that our model is competitive or state of the art on several classification tasks, while trailing on speaker-related tasks.

1 Introduction

Digital representations of audio signals are focused on recreation of the source audio and do not carry rich information about the content of the audio with them. Historically, transforms such as Mel-Frequency Cepstrum Coefficients (MFCC) and Discrete Cosine Transforms (DCT), and later deep neural networks, have been used to extract meaningful information from the Audio to perform discriminative tasks. While the former two examples are generalized transforms, they are the first step in task-specific pipelines, while the latter example is often trained from scratch on a specific task. General Purpose Audio Embeddings (GPAE) seeks to find low-dimensional, rich representations which can be used on several, disparate downstream tasks.

Recent years have seen many approaches to pretraining for embeddings to extract semantic and qualitative information from audio. Benchmarks, such as SUPERB (Yang et al. [2021]) and HEAR (Turian et al. [2022]) have been introduced to test proposed embeddings on several tasks, creating a unified framework to compare different approaches.

In our work, we seek GPAE based on the work of Schmid et al. [2023b] as a preliminary study as to the applicability of Neural Codecs (Zeghidour et al. [2021], Défossez et al. [2022]) to this task. While we do not enjoy state-of-the-art results on any tasks, we find competitive results on a few tasks and reasonable results on many other tasks.

2 Related Work

Audio Benchmarks. SUPERB (Yang et al. [2021]) is a benchmark released in 2021 as a set of disparate tasks related to speech processing. The goal of SUPERB is to compare the performance of large self-supervised speech models. Common tasks in Superb include Automatic Speaker Verification (ASV), Speaker Identification (SID), and Automatic Speech Recognition (ASR). HEAR (Turian et al. [2022]) is a similar benchmark, though it includes tasks from music information retrieval and environmental sound backgrounds. Moreover, HEAR is more focused on embedding representations, whereas SUPERB is meant to evaluate models themselves. In this work, we primarily target HEAR tasks, though we perform some light analysis on SUPERB tasks. HARES (Wang et al. [2021]) is a benchmark proposed by DeepMind which sits as a middle ground between the others, with more Speech than HEAR but more General Audio than SUPERB.

Preprint. Under review.

Audio Models for Speech Representations. The two most common models for extracting low-dimension representations of audio are HuBERT (Hsu et al. [2021]) and wav2vec (Schneider et al. [2019], Baevski et al. [2020]). The former learns an audio and language model with a BERT-like masked objective. The learned representations are then combined in a k-means clustering step. The latter uses an encoder combined with a context network and quantization module to create discrete audio tokens.

Audio Models for Music Representations. Music representations differ from speech since Music typically has much longer time-specific structure in any given example. As such, music-specific models are trianed to create musical representations. CREPE (Kim et al. [2018]) is a model which uses convolutional layers to provide pitch estimation. MuLan (Huang et al. [2022]) uses contrastive pretraining to create music and language embeddings. COALA (Favory et al. [2020]) seeks audio representations by training two, aligned autoencoders on the time-frequency representation of audio and the semantic information of tags. MusCALL (Manco et al. [2022]) follows a similar approach, but does not require downstream fine-tuning like COALA.

General Purpose Audio Models. CLAP (Manco et al. [2022]) and Wav2CLIP (Wu et al. [2022a]) both use contrastive loss to produce embeddings. The former applies CLIP's (Radford et al. [2021]) methodology while swapping images and audios, while the latter distills from CLIP. COLA (Saeed et al. [2021]) trains a contrastive model on AudioSet (Gemmeke et al. [2017]) while targeting music, speech, and environmental audio. Dinkel et al. [2023] proposes a method called consistent teaching which combines input augmentation and knowledge distillation to train a vision transformer on audio tagging tasks. Most related to our work is the work of Schmid et al. [2023b] which uses knowledge distillation to train a MobileNet on an audio tagging task. Layers are then extracted as embeddings which have shown good results on the HEAR benchmark.

3 Methodology

We follow the approach of Schmid et al. [2023b] which extracts GPAE from prior work which trained a MobileNet (Howard et al. [2019]) using knowledge distillation on MFCCs (Schmid et al. [2023a]). We use this approach with few modifications, most notably replacing MFCCs with Encodec (Défossez et al. [2022]) tokens and experimenting with the learning rate scheduling.

We begin with preprocessing SALT lab's collection of AudioSet (SALT-AS) (Gemmeke et al. [2017]) for offline knowledge distillation. While Schmid et al. [2023b] uses an ensemble of PaSST (Koutini et al. [2022]), we are limited by time and compute and instead opt for a single pretrained Audio Spectrogram Transformer (AST, Gong et al. [2021]) as our teacher model. We use the monophonic, 24 kHz version of EnCodec with a target bitrate of 24 kbps.

During pre-training, we first retrieve the quantized vectors of the EnCodec model and freeze this layer to avoid updating any weights of the EnCodec model. The resulting matrix has dimensions of 750×128. We use a CNN-based transform to project the tokens to the expected input size of the MobileNet of $3 \times 224 \times 224$. As described in Schmid et al. [2023a], we add parameter α to the MobileNet which is a scaling parameter to change the width of each Block in the MobileNet.

We train each pretrained model for 200 epochs, where each epoch consists of 100,000 examples drawn from an unbalanced subset of SALT-AS with approximately 1.7M examples. We form a validation set from the SALT-AS balanced subset. We use the `OneCycleLR` scheduler provided by PyTorch with cosine annealing, warming up over 8 epochs to a maximum learning rate of 8×10^{-4}. For knowledge distillation, we fix the temperature $\tau = 1.0$ and the teacher weight, β to be either 0.9 or 0.0, where the latter forms a baseline of a MobileNet trained only with binary cross entropy classification loss.

$$\mathcal{L} = (1 - \beta)\mathcal{L}_{\text{BCE}} + \beta\mathcal{L}_{\text{KD}} \tag{1}$$

We then take our best performing pretrained model and extract GPAEs from it by using the same method that Schmid et al. [2023b] uses for their mid-level feature embeddings. Namely, we extract embeddings from blocks 5, 11, 13, and 15 of the MobileNet model, we do an average pool over the time dimensions, and we concatenate the resulting feature values. We chose the mid-level features over the high-level features, the low-level features, and the squeeze-and-excitation features because it

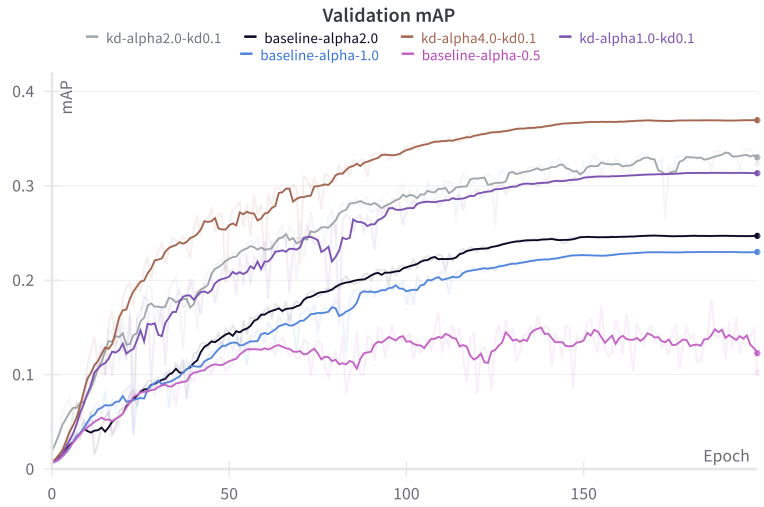

Figure 1: Pretraining results. Validation mAP for various pretrained models with both knowledge distillation and without. Baseline models are trained with only binary cross entropy loss, while kd models are trained with knowledge distillation with $\tau = 1.0$ and $\beta = 0.9$. The presented lines are smoothed by a moving average of 6 epochs with the un-smoothed line translucent behind the smoothed line. We do not report results on a baseline where $\alpha = 4.0$ as training stopped early.

was the single set of features that performed the best overall in the HEAR benchmark in Schmid et al. [2023b]. We used these embeddings for training and evaluating the downstream benchmark tasks.

4 Results

In this section we discuss our results for both pretraining and a few benchmarks from both HEAR Turian et al. [2022] and SUPERB Yang et al. [2021].

4.1 Pre-Training

For pretraining, we perform limited parameter search on models by selecting $\alpha = \{0.5, 1.0, 2.0, 4.0\}$, $\beta = \{0.0, 0.9\}$, and $\tau = 1.0$. We choose these values for β and τ as they are the best performing hyperparameter settings in Schmid et al. [2023a]. Our best performing model reaches a validation mAP of 0.370, indicating decent performance in comparison to historic approaches to audio tagging using AudioSet. This result, however, is far lower than the findings in Schmid et al. [2023a], which reports a mAP of 0.483.

Unsurprisingly, the best performing model is also the largest, where $\alpha = 4.0$. For all of the pretrained models, we see increasing the α bolsters the mAP, as expected. Moreover, knowledge distillation provides a large boost to the mAP across all model sizes. For example in the model where $\alpha = 2.0$, the baseline reaches a mAP 0.247, while with knowledge distillation, the model is able to achieve 0.330.

4.2 Benchmarks

Table 1 shows our results compared to those of other models on ten of the HEAR benchmark tasks. Our model seems to perform best on the NSynth Pitch and Speech Commands tasks. NSynth Pitch 50h is the only task where our model matches the best-performing model. However, our model does outperform the simple wav2vec2 and Hubert baselines on a majority of the tasks. (CREMA-D and VoxLingua-107 are the only two tasks in which our model performs worse than both wav2vec2 and GURA Hubert.)

Table 1: Results on the HEAR benchmark versus other models. We compare against HuBERT and wav2vec2.0 baselines, as well as the results from Schmid et al. [2023b] (mn40_as) and Dinkel et al. [2023] (ced_base). HuBERT Fuse Cat (Wu et al. [2022b] fuses and concatenates embeddings from HuBERT, wav2vec2.0, and CREPE.

Models	CREMA-D	ESC-50	GTZAN Genre	Libricount	Vocal Imitation
Ours	0.5367	0.6565	0.8320	0.6640	0.1591
mn40_as	0.6400	0.9615	**0.9080**	**0.7253**	0.2019
GURA Fuse Cat	**0.7474**	0.7335	0.8050	0.6972	0.1969
ced_base	0.6910	**0.9665**	0.8860	0.6785	**0.2269**
wav2vec2	0.6562	0.5610	0.7800	0.6921	0.08006
GURA Hubert	0.6897	0.6035	0.7350	0.6458	0.1540
Models	VoxLingua-107 top 10	NSynth Pitch 5h	NSynth Pitch 50h	Speech Commands 5h	Speech Commands Full
Ours	0.2902	0.7720	**0.8852**	0.8744	0.9577
mn40_as	0.3179	0.3040	0.6355	0.7767	0.8472
GURA Fuse Cat	**0.7202**	**0.8460**	**0.8852**	**0.9605**	**0.9677**
ced_base	0.3857	0.6820	0.8281	0.8693	0.8967
wav2vec2	0.4928	0.4020	0.6530	0.8382	0.8785
GURA Hubert	0.6368	0.1840	0.4288	0.9530	0.9540

Table 2: Results on the SUPERB benchmark versus other models. We compare against $HuBERT - Base$ for a standard comparison, and $DistilHuBERT$ for a low-parameter baseline.

Models	Params	Keyword Spotting 1	Speaker Identification (SID)	Emotion Recognition (ER)
Ours	15M	0.78	0.19	0.56
HuBERT Hsu et al. [2021]	96M	**0.96**	**0.81**	**0.65**
DistilHuBERT Chang et al. [2021]	24M	0.96	0.74	0.62

Our model mn40_as is of special interest, since it is a version of the MobileNet model that our model is based off of. The main differences between our model and mn40_as are that our MobileNet model uses EnCodec tokens instead of mel spectrogram features as input, our model is pretrained by distilling from a single model instead of a transformer ensemble, and our model uses a simpler embedding extraction strategy. Our model significantly outperforms mn40_as in all four NSynth Pitch and Speech Commands tasks, while mn40_as outperforms our model on all others.

Table 2 demonstrates the results of our model on SUBERB compared to baseline models with different parameter counts. We choose to evaluate on keyword spotting to test for word recognition in a mean-pooled setting, and emotion recognition to test for classification-related results on speech. We find that our model largely underperforms on these speech-related tasks, particularly in SID. This conforms with AS data, where speaker information is not predictive of downstream labels. This is somewhat surprising with respect to EnCodec units, however, as papers like VALL-E (Wang et al. [2023]). This indicates that a drastic difference in parameter count may be causing less observable EnCodec features to be difficult for the model to analyze.

5 Discussion

Overall, we find that our results largely underperform against the baseline models in speaker tasks. The lower performance largely comes in speaker-related tasks, such as low scores in speaker identification (SID) from SUPERB and LibriCount from HEAR. The changes our model introduces largely come from two places – a full pretrain on AudioSet and the introduction of EnCodec codes. The domain mismatch from AudioSet pretraining has been observed in prior work, such as the AST pretrained on AudioSet classification (Gong et al. [2021]) underperforming comparatively on SID with a score

of 35.2 while getting close-to-SOTA scores on tasks like KS and ER. Using EnCodec codes has also been observed to be lower-quality for language pretraining, such as in AudioLM (Borsos et al. [2023]).

However, on classification tasks, our model performs surprisingly well, achieving high-quality results on GTZAN Genre and NSynth Pitch 50h. This fits a real-world scenario where EnCodec units are streamed as compressed data to a phone and during decoding alternative classification tasks can be applied. Future work in this direction can focus on training lightweight classification heads for audio tasks specific to a downstream user, allowing privacy towards preferences while maintaining quality and speed.

6 Conclusion

We introduce a model for classification using general purpose audio embeddings from EnCodec units. We train using a MobileNet architecture and see competitive performance in several classes of tasks, particularly in music-related classification, with a 5x reduced parameter count compared to HuBERT. We hope that these techniques will help toward the goal of having private and on-device machine learning algorithms that can customize to their users' preferences.

References

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, October 2020.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. Audiolm: a language modeling approach to audio generation, 2023.

Heng-Jui Chang, Shu-Wen Yang, and Hung-yi Lee. Distilhubert: Speech representation learning by layer-wise distillation of hidden-unit BERT. *CoRR*, abs/2110.01900, 2021. URL `https://arxiv.org/abs/2110.01900`.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High Fidelity Neural Audio Compression, October 2022.

Heinrich Dinkel, Yongqing Wang, Zhiyong Yan, Junbo Zhang, and Yujun Wang. CED: Consistent ensemble distillation for audio tagging, September 2023.

Xavier Favory, Konstantinos Drossos, Tuomas Virtanen, and Xavier Serra. COALA: Co-Aligned Autoencoders for Learning Semantically Enriched Audio Representations. June 2020.

Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio Set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, New Orleans, LA, March 2017. IEEE. ISBN 978-1-5090-4117-6. doi: 10.1109/ICASSP.2017.7952261.

Yuan Gong, Yu-An Chung, and James Glass. AST: Audio Spectrogram Transformer, July 2021.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for MobileNetV3, November 2019.

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, June 2021.

Qingqing Huang, Aren Jansen, Joonseok Lee, Ravi Ganti, Judith Yue Li, and Daniel P. W. Ellis. MuLan: A Joint Embedding of Music Audio and Natural Language, August 2022.

Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. CREPE: A Convolutional Representation for Pitch Estimation, February 2018.

Khaled Koutini, Jan Schlüter, Hamid Eghbal-zadeh, and Gerhard Widmer. Efficient Training of Audio Transformers with Patchout. In *Interspeech 2022*, pages 2753–2757, September 2022. doi: 10.21437/Interspeech.2022-227.

Ilaria Manco, Emmanouil Benetos, Elio Quinton, and György Fazekas. Contrastive Audio-Language Learning for Music, August 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision, February 2021.

Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive Learning of General-Purpose Audio Representations. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3875–3879, June 2021. doi: 10.1109/ICASSP39728.2021.9413528.

Florian Schmid, Khaled Koutini, and Gerhard Widmer. Efficient Large-scale Audio Tagging via Transformer-to-CNN Knowledge Distillation, June 2023a.

Florian Schmid, Khaled Koutini, and Gerhard Widmer. Low-Complexity Audio Embedding Extractors, June 2023b.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. Wav2vec: Unsupervised Pre-training for Speech Recognition, September 2019.

Joseph Turian, Jordie Shier, Humair Raj Khan, Bhiksha Raj, Björn W. Schuller, Christian J. Steinmetz, Colin Malloy, George Tzanetakis, Gissel Velarde, Kirk McNally, Max Henry, Nicolas Pinto, Camille Noufi, Christian Clough, Dorien Herremans, Eduardo Fonseca, Jesse Engel, Justin Salamon, Philippe Esling, Pranay Manocha, Shinji Watanabe, Zeyu Jin, and Yonatan Bisk. HEAR: Holistic Evaluation of Audio Representations, May 2022.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. Neural codec language models are zero-shot text to speech synthesizers, 2023.

Luyu Wang, Pauline Luc, Yan Wu, Adria Recasens, Lucas Smaira, Andrew Brock, Andrew Jaegle, Jean-Baptiste Alayrac, Sander Dieleman, Joao Carreira, and Aaron van den Oord. Towards Learning Universal Audio Representations. https://arxiv.org/abs/2111.12124v3, November 2021.

Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. Wav2CLIP: Learning Robust Audio Representations from Clip. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4563–4567, May 2022a. doi: 10.1109/ICASSP43922.2022.9747669.

Tung-Yu Wu, Chen-An Li, Tzu-Han Lin, Tsu-Yuan Hsu, and Hung-Yi Lee. The Efficacy of Self-Supervised Speech Models for Audio Representations. https://arxiv.org/abs/2209.12900v3, September 2022b.

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, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko-tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee. SUPERB: Speech processing Universal PERformance Benchmark, October 2021.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. SoundStream: An End-to-End Neural Audio Codec, July 2021.