CED: CONSISTENT ENSEMBLE DISTILLATION FOR AUDIO TAGGING

Heinrich Dinkel, Yongqing Wang, Zhiyong Yan, Junbo Zhang, Yujun Wang

Xiaomi Corporation, Beijing, China

ABSTRACT

Augmentation and knowledge distillation (KD) are well-established techniques employed in audio classification tasks, aimed at enhancing performance and reducing model sizes on the widely recognized Audioset (AS) benchmark. Although both techniques are effective individually, their combined use, called consistent teaching, hasn't been explored before. This paper proposes CED, a simple training framework that distils student models from large teacher ensembles with consistent teaching. To achieve this, CED efficiently stores logits as well as the augmentation methods on disk, making it scalable to large-scale datasets. Central to CED's efficacy is its label-free nature, meaning that only the stored logits are used for the optimization of a student model only requiring 0.3% additional disk space for AS. The study trains various transformer-based models, including a 10M parameter model achieving a 49.0 mean average precision (mAP) on AS. Pretrained models and code are available online.

Index Terms— audio tagging, audio classification, efficient data storage, teacher-student, knowledge distillation.

1. INTRODUCTION

Audio tagging (AT) is a task that categorizes sounds into a fixed set of event classes, e.g., a baby crying or water running. Applications of AT systems include aid for the hearing impaired, general monitoring of sounds [1, 2] as well as additional targets for keyword spotting [3, 4]. Enhancing performance and minimizing the size of AT systems is vital for practical deployment. We target performance and size enhancement through common methods: data augmentation and knowledge distillation (KD).

In KD, a large teacher model generates soft labels (logits) for a smaller student model to learn from. Typically, the objective of KD involves optimizing both the original hard labels and the logits together. Yet, recent research [7] found that using only logits as training targets can significantly improve performance compared to the usual method. By combining KD and data augmentation, also known as consistent teaching [12], it has been suggested that performance can be further boosted. Surprisingly, no previous research has applied this approach to AT. We believe that the limited exploration of this approach is due to challenges in efficiently implementing KD.
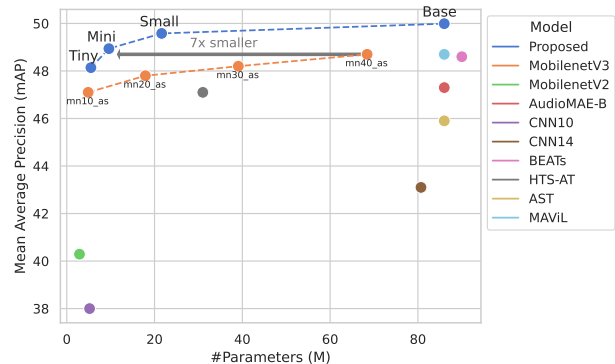

Fig. 1. Our achieved performance in comparison to other works on Audioset (AS-2M). We reference results from the works in [5, 6, 7, 8, 9, 10, 11].

Practically, there are two main ways to implement KD: 1. Online KD infers each soft label during training by forwarding a sample jointly through the teacher as well as the student. 2. Offline KD stores augmented samples as well as the teacher's soft labels on disk and reads them during student training. Both these methods have pros and cons, detailed in Figure 2. Online KD is handicapped by its slow training speed because samples need to be sequentially forwarded through student and teacher, while offline KD can "parallelize" this process by first creating the teacher's data/logits. Conversely, offline KD struggles when handling substantial augmented data due to the storage demand of augmented samples. Thus, in practice, only logits from non-augmented training data are stored on disk. Moreover, the performance of offline KD drops when (inconsistent) data augmentation techniques are used on the student's input, as demonstrated in previous studies [7, 13]. Our research shows (Section 4.1) that consistent augmentation for both teacher and student inputs is crucial to improve performance.

We would like highlight key distinctions between our research two comparable works, Efficient-AT [13] and PSL [7] works. Efficient-AT [13] has utilized standard offline KD to achieve state-of-the-art (SOTA) performance on Audioset [14] (AS), but due to the nature of offline KD, could not apply consistent teaching. Further, work in [7] showed that label-free online KD is feasible for AT, yet could only use simple teacher models (MobileNetV2), since larger mod-
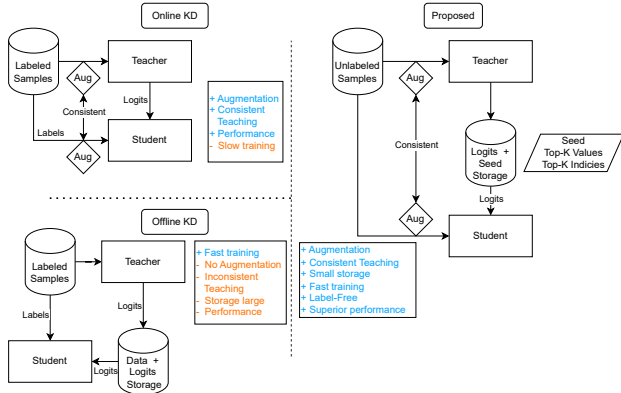
Fig. 2. CED in comparison to standard on/off-line KD frameworks. We first augment each sample using wave-level and spectrogram-level augmentations. Then we use an (ensemble) teacher model to predict scores for each respective sample and only store the seed which generated the augmentations and the top-k scores on disk.

els significantly slow down training. However, to the best of our knowledge, there has been no previous work that combined KD and augmentation with label-free online KD. Finally, this work is closely related to the computer vision work TinyViT [15], which applied a similar training pipeline for image classification. Our contributions are as follows: (I) We propose CED, a simple framework to efficiently store and access logits as well as augmentation methods. CED only requires a few bytes of storage per sample, making it scalable to large datasets. (II) We introduce consistent teaching to AT, which improves performance and reduces the performance gap between teacher and student models. (III) We show that the features of CED models are also transferable to other audio classification tasks.

2. METHOD

A trivial solution to enable consistent teaching is to store an entire augmented dataset for each epoch on disk, use a teacher to predict logits and save those logits. However, this solution is impractical for sizeable datasets, such as AS, due to its considerable storage demand (500 GB per epoch). Further, audio can be augmented on wave and spectrogram levels, meaning that one would need to save augmented waveforms as well as their respective spectrograms, further increasing the storage requirement.

Instead of storing augmented samples on disk, CED only stores the seed that generated each respective augmented sample. Specifically, given a single audio sample $\mathbf{x} \in \mathbb{R}^{T_w}$ with T_w taps, we first apply wave-level augmentations $\hat{\mathbf{x}} = \text{wavaug}(\mathbf{x}, \phi)$ using a random seed ϕ on the sample. Then, a Mel-spectrogram $\mathbf{X} \in \mathbb{R}^{F \times T}$ is extracted from $\hat{\mathbf{x}}$ and augmented using spectrogram-level augmenta-

tions: $\hat{\mathbf{X}} = \text{specaug}(\mathbf{X}, \phi)$. Instead of storing $\hat{\mathbf{x}}$ and $\hat{\mathbf{X}}$, we efficiently store ϕ on disk for each sample.

Then we use a teacher model $\mathcal{T}$ to predict logits $\mathbf{y}_T \in [0, 1]^C$ from the given sample $\hat{\mathbf{X}}$ for C classes during each epoch $e = 1, \ldots, E$. Directly storing $\mathbf{y}_T$ for a dataset with N samples, C classes and E epochs requires NCE disk space. In the case of AS, $N = 2 \times 10^6, C = 527$, results in approximately 4 GB (float 32) of storage per epoch.

In our study, we've chosen to preserve only the most prominent K logits $\hat{\mathbf{y}}_{top-k}$ for each sample, along with the corresponding label indices $\hat{\mathbf{y}}_{top-idx}$. Overall, we save $\{\hat{\mathbf{y}}_{top-k}, \hat{\mathbf{y}}_{top-idx}, \phi\}$ as 16-bit float, 16-bit integer and 32-bit integer, respectively. This approach leads to a storage requirement of $((K \times (2 \times 2)) + 4)$ bytes per logit. For our chosen $K = 20$, this results in approximately 84 bytes, or around 80 MB per AS-2M epoch. Notably, this is significantly lower compared to the naïve solution that would demand 4 GB. As we only retain K logits on disk, we assume a remaining probability of 0 for each sample. While we investigated alternative methods for handling remaining probabilities, these methods did not yield noticeable improvements.

3. EXPERIMENTS

3.1. Dataset

Our training and evaluation dataset is AS [14], which mainly contains 10-second-long audio clips labelled with 527 different sound event classes. We collected 1,904,746 training samples and 18,299 evaluation samples sampled at 16 kHz. The training set is split into two subsets, the balanced (AS-20K) subset with 21,155 samples, and the entire full (AS-2M) subset with 1.9M samples. Our experiments are first run on the AS-20K subset to ascertain our method's effect and then the findings are applied by training on AS-2M.

3.2. Models

Student models The present study employs four vision transformer (ViT)-based architectures (see Table 1), namely (CED-) Tiny, Mini, Small, and Base [16], all of which closely follow ViT (pre-norm + GeLU activation). Each model's setup follows [2], where we employ 64-dimensional banks at a 16 kHz sampling rate, extracted within a 32 ms window and a 10 ms hop. These filterbanks are first normalized using batch normalization. From these spectrograms, we extract non-overlapping patches with a size of 16×16, resulting in $252 = 62 \times 4$ patches for a 10 s input in time/frequency, respectively. We use absolute positional embeddings, where time and frequency are independently modeled to allow these models to handle variable-sized inputs during inference. In order to achieve competitive performance [13], we use the masked autoencoder paradigm (MAE) to pre-train all teacher and student models [11, 2] on AS.

Model	# Parameter	Embed	MLP	#Heads
Tiny	5.5 M	192	768	3
Mini	10 M	256	1024	4
Small	22 M	384	1536	6
Base	86 M	768	3072	12

Table 1. The utilized models in this study. "Embed" refers to the embedding dimension, "MLP" to the dimension of each block's multi-layer perceptron and "#Heads" stands for the number of independent attention mechanisms.

Teacher model Inspired by [13], we use an ensemble of differently-sized transformer models to predict labels. Specifically, the ensemble consists of two ViT-Base models and three ViT-Large models, which all have been independently trained on AS. Since CED requires consistency between the input features, the Mel-spectrogram configuration of each teacher is identical to a student. The ensemble teacher model achieves an mAP of 50.1 on the evaluation set of AS.

3.3. Augmentations and Logits

Even though our pipeline can support a plethora of augmentation methods, we keep the augmentations in this work simple. In the waveform domain, we use sample-level shifting of the signal. Further, in the spectrogram domain, we use SpecAug [17], masking at most 192 time-frames and 24 frequency banks. On AS-20K we additionally apply mixup [18] with $\lambda = 0.5$. In order to further conserve storage space, we only save a certain amount of epochs and cycle over the dataset during training, which we set to $E = 40$ for AS-20K and $E = 10$ for AS-2M. Further, we use $K = 20$ as the default for all experiments, leading to an overall size of 1.5 GB, or 0.3% of the AS-2M data. A storage requirement comparison between CED and previous works can be seen in Table 2.

Method	Aug?	AS-20K	AS-2M
Naïve	✗	42	3800
Efficient-AT [13]	✗	21	1900
Proposed ($K = 20$)	✓	**1.8**	**155**

Table 2. Logit storage requirement per epoch given in Megabytes (MB). "Naïve" refers to storing all $C = 527$ logits using float32 precision and "Efficient-AT" uses float16 precision. "Aug?" indicates consistent augmentation support.

3.4. Setup

The majority of works on AS-2M use a balanced sampling strategy [10] due to its long-tailed label distribution, which has been shown to have a negative impact on other datasets [19]. In line with the label-free nature of our work,

$\mathcal{T}_{Aug}$	$\mathcal{S}_{Aug}$	Tiny	Mini	Small	Base
✗	✗	28.52	30.52	32.28	37.87
✗	✓	28.77	30.35	32.54	37.09
✓	✗	31.75	33.45	34.30	39.03
✓	✓	36.47	38.50	41.55	43.97

Table 3. Impact of consistent training between teacher and student. The training data is AS-20K and values represent mAP on the AS evaluation set, where higher is better.

we *do not use a balanced sampling strategy*, since our method has no access to hard labels during training and thus sample randomly. We train with an 8-bit Adam optimizer [20] using a cosine learning rate decay scheduler and a maximal learning rate of 0.001 for the Tiny/Mini models and 0.0003 for the Small/Base models. We warmup the learning rate for 5,000 and 62,500 batches for AS-20K and AS-2M, respectively and decay the learning rates to 10% of their maximal value over the training period. We use the standard binary cross entropy loss (BCE) between the student's predicted logits and the teacher's logits as the training objective and the main evaluation metric is the mean average precision (mAP). Training runs for 300 epochs with a batch size of 32 on AS-20K and for 120 epochs with a batch size of 128 on AS-2M. Overall, logit extraction takes 30 hours and training takes at most 4 days on a single A100 GPU, depending on the model size. We use AS-20K for model analysis and ablation studies. The neural network back-end is implemented in Pytorch [21] and the source code with pretrained checkpoints is publicly available[1].

4. RESULTS

4.1. Consistent teaching

Here, we provide evidence that using consistent augmentations between teacher and student is crucial to improving performance. The results of our experiment can be seen in Table 3, where $\mathcal{T}_{Aug}$ represents using augmentation during logit prediction and $\mathcal{S}_{Aug}$ represents applying augmentation (see Section 3.3) on the student's input. In summary, introducing augmentation to student inputs produces minor performance changes (within -0.8 to +0.2 mAP) across models, in line with [13]. Conversely, augmenting only teacher inputs yields noteworthy enhancements, surpassing the baseline by more than 2 mAP points. Finally, when using CED, which applies consistent teacher-student augmentation, substantial performance improvements emerge. These gains range from 5 to 7 mAP points over the baseline.
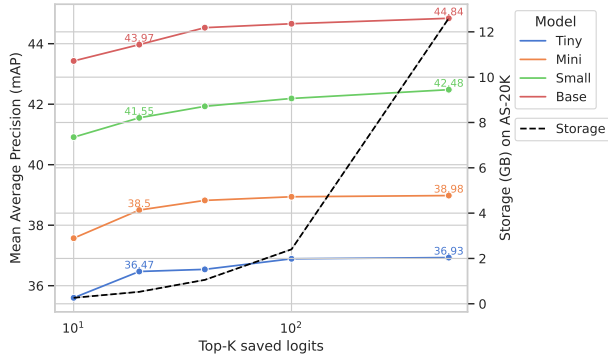
---

[1] https://github.com/RicherMans/ced

Fig. 3. Performance and storage impact of different K in AS-20K. We depict the achieved mAP for $K = 20$ (proposed) and $K = 527$ (best). Best viewed in color.

4.2. Impact of saving K logits

Figure 3 presents our findings on the AS-20K dataset using $K = \{10, 20, 40, 100, 527\}$, while setting $E = 300$. Generally, larger K correlates with improved performance, though this improvement comes at the cost of heightened storage needs. The performance gap between $K = 10$ and $K = 20$ (proposed) is significant, showing a 1-point mAP difference, while the storage impact remains minimal. Note that for $K = 527$, the storage demand increases to 12 GB, an additional 190% of the original dataset size of 6.3 GB. In summary, the value of K can be customized according to the dataset size and storage availability, given its direct influence on both performance and storage demands.

4.3. Main results

The main results of CED when compared to previous work can be seen in Table 4. Notably, our Mini model showcases superior performance, achieving an mAP of 49.0, while utilizing a mere 10 million parameters. Moreover, our single Base model with an mAP of 50 exhibits only a slight performance gap in comparison to the 5-way ensemble teacher with 50.1.

4.4. Transfer to downstream tasks

Here we investigate whether CED-trained features are transferable to other downstream tasks, specifically for sound event detection (FSD50K, DCASE16) and acoustic scene classification (ESC-50). To assess this, we employ the HEAR [24] benchmark, which employs a linear classifier atop extracted features. For all experiments, we extract features from the penultimate layer of our model by mean averaging all patches. Results can be seen in Table 5, where CED-trained models are compared against alternative AS-based approaches. CED-trained models can be seen to perform well across a variety of sound-related downstream tasks.

---

[2] On our evaluation split the model obtains 46.6.

	Model	#Par (M)	AS-20K	AS-2M
Baseline	CNN14 [10]	81	27.8	43.1
	MobileNetV2 [7]	2.9	35.5	40.3
	HTS-AT [8]	31	-	47.1
	AST [6]	86	34.7	45.9
	MaskSpec [22]	86	32.3	47.1
	BEATs [5]	90	38.9	48.6[2]
	AudioMAE-B [11]	86	37.0	47.3
	ConvNeXt [23]	28	-	47.1
	MN10-AS [13]	4.9	-	47.1
	MN20-AS [13]	18	-	47.8
	MN40-AS [13]	68	-	48.7
	MAViL [9]	86	41.8	48.7
CED	Tiny	5.5	36.5	48.1
	Mini	10	38.5	49.0
	Small	22	41.6	49.6
	Base	86	**44.0**	**50.0**

Table 4. Main results on AS-20K and AS-2M. Models highlighted in grey have been trained with multi-modal supervision (Audio + Visual). Best in bold.

Model	FSD50K	ESC-50	DCASE16
CNN14	-	90.85	0.0
Eff-B2	60.71	93.45	79.01
PaSST	64.09	94.75	78.79
MN-40AS	63.12	96.15	81.30
Tiny	62.73	95.80	88.02
Mini	63.88	95.35	90.66
Small	64.33	95.95	91.63
Base	**65.48**	**96.65**	**92.19**

Table 5. Linear evaluation results on the HEAR benchmark for sound detection tasks in comparison to previous works. Additional results are publicly available. Best in bold.

5. CONCLUSION

This work introduced CED, a simple training framework for distilling AT models with consistent teaching. Our work aims to efficiently distil a single model from an ensemble of large teacher models by storing the teacher model's logits as well as their respective augmentation method on disk. Our results show that with CED, we can efficiently distil single models that are capable of achieving performance similar to large ensembles. The Mini network can achieve an mAP of 49.0, outperforming previous studies by a significant margin, with a fraction of the number of parameters. While this work focuses on transformer-based teacher and student models, it is important to note that CED is a general framework and can be used to distil other network types.

6. REFERENCES

[1] Wuyue Xiong, Xuenan Xu, Long Chen, and Jian Yang, "Sound-based construction activity monitoring with deep learning," *Buildings*, vol. 12, no. 11, pp. 1947, Nov 2022.

[2] Heinrich Dinkel, Zhiyong Yan, Yongqing Wang, Junbo Zhang, and Yujun Wang, "Streaming audio transformers for online audio tagging," 2023.

[3] Heinrich Dinkel, Yongqing Wang, Zhiyong Yan, Junbo Zhang, and Yujun Wang, "UniKW-AT: Unified Keyword Spotting and Audio Tagging," in *Proc. Interspeech 2022*, 2022, pp. 3238–3242.

[4] Heinrich Dinkel, Yongqing Wang, Zhiyong Yan, Junbo Zhang, and Yujun Wang, "Unified keyword spotting and audio tagging on mobile devices with transformers," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[5] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, and Furu Wei, "Beats: Audio pretraining with acoustic tokenizers," 2022.

[6] Yuan Gong, Yu-An Chung, and James Glass, "AST: Audio Spectrogram Transformer," in *Proc. Interspeech 2021*, 2021, pp. 571–575.

[7] Heinrich Dinkel, Zhiyong Yan, Yongqing Wang, Junbo Zhang, and Yujun Wang, "Pseudo strong labels for large scale weakly supervised audio tagging," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 336–340.

[8] Ke Chen, Xingjian Du, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov, "Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 646–650.

[9] Po-Yao Huang, Vasu Sharma, Hu Xu, Chaitanya Ryali, Haoqi Fan, Yanghao Li, Shang-Wen Li, Gargi Ghosh, Jitendra Malik, and Christoph Feichtenhofer, "Mavil: Masked audio-video learners," *arXiv preprint arXiv:2212.08071*, 2022.

[10] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[11] Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer, "Masked autoencoders that listen," *Advances in Neural Information Processing Systems*, vol. 35, pp. 28708–28720, 2022.

[12] Lucas Beyer, Xiaohua Zhai, Amélie Royer, Larisa Markeeva, Rohan Anil, and Alexander Kolesnikov, "Knowledge distillation: A good teacher is patient and consistent," 2022.

[13] Florian Schmid, Khaled Koutini, and Gerhard Widmer, "Efficient large-scale audio tagging via transformer-to-cnn knowledge distillation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[14] Jort F Gemmeke, Daniel PW Ellis, Dan Freedman, Albert Jansen, Mike Lawrence, R. C. Moore, Manoj Plakal, Matt Ritter, Malcolm Slaney, Ron J Weiss, et al., "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[15] Kan Wu, Jinnian Zhang, Houwen Peng, Mengchen Liu, Bin Xiao, Jianlong Fu, and Lu Yuan, "Tinyvit: Fast pretraining distillation for small vision transformers," in *European Conference on Computer Vision*. Springer, 2022, pp. 68–85.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021.

[17] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[18] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[19] R. Channing Moore, Daniel P. W. Ellis, Eduardo Fonseca, Shawn Hershey, Aren Jansen, and Manoj Plakal, "Dataset balancing can hurt model performance," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[20] Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer, "8-bit optimizers via block-wise quantization," *9th International Conference on Learning Representations, ICLR*, 2022.

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala, "PyTorch: An imperative style, high-performance deep learning library," vol. 32, pp. 8026–8037. Curran Associates, Inc., 2019.

[22] Dading Chong, Helin Wang, Peilin Zhou, and Qingcheng Zeng, "Masked spectrogram prediction for self-supervised audio pre-training," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[23] Thomas Pellegrini, Ismail Khalfaoui-Hassani, Etienne Labbé, and Timothée Masquelier, "Adapting a convnext model to audio classification on audioset," *arXiv preprint arXiv:2306.00830*, 2023.

[24] Joseph Turian, Jordie Shier, Humair Raj Khan, Bhiksha Raj, Björn W Schuller, Christian J Steinmetz, Colin Malloy, George Tzanetakis, Gissel Velarde, Kirk McNally, et al., "Hear: Holistic evaluation of audio representations," in *NeurIPS 2021 Competitions and Demonstrations Track*. PMLR, 2022, pp. 125–145.