uaMix-MAE: EFFICIENT TUNING OF PRETRAINED AUDIO TRANSFORMERS WITH UNSUPERVISED AUDIO MIXTURES

Afrina Tabassum[1,*], *Dung Tran*[2], *Trung Dang*[2], *Ismini Lourentzou*[1,3], *Kazuhito Koishida*[2]

[1] Virginia Tech, [2]Applied Sciences Group, Microsoft Corporation
[3]University of Illinois at Urbana - Champaign

ABSTRACT

Masked Autoencoders (MAEs) learn rich low-level representations from unlabeled data but require substantial labeled data to effectively adapt to downstream tasks. Conversely, Instance Discrimination (ID) emphasizes high-level semantics, offering a potential solution to alleviate annotation requirements in MAEs. Although combining these two approaches can address downstream tasks with limited labeled data, naively integrating ID into MAEs leads to extended training times and high computational costs. To address this challenge, we introduce uaMix-MAE, an efficient ID tuning strategy that leverages unsupervised audio mixtures. Utilizing contrastive tuning, uaMix-MAE aligns the representations of pretrained MAEs, thereby facilitating effective adaptation to task-specific semantics. To optimize the model with small amounts of unlabeled data, we propose an audio mixing technique that manipulates audio samples in both input and virtual label spaces. Experiments in low/few-shot settings demonstrate that uaMix-MAE achieves $4 - 6\%$ accuracy improvements over various benchmarks when tuned with limited unlabeled data, such as AudioSet-20K. Code is available at https://github.com/PLAN-Lab/uamix-MAE

Index Terms— Masked audio models, Contrastive tuning, Few-shot learning, Masked autoencoders

1. INTRODUCTION

Self-supervised learning has attracted significant attention for its ability to learn meaningful representations from vast amounts of unlabeled data, mitigating the need for costly annotations. Besides significant advancements in computer vision [1, 2, 3] and natural language processing [4, 5, 6], self-supervised learning has also recently demonstrated potential for various speech and audio understanding tasks [7, 8, 9, 10, 11, 12]. Two highly effective self-supervised techniques in speech and audio understanding are Masked Audio Modeling (MAM) [7, 8, 9, 13] exemplified by methods such as MAE [7], and Instance Discrimination (ID) [10, 11, 12, 14].

MAE [3] employs a pre-training task where audio inputs are partitioned into non-overlapping patches, and a subset of

---

these patches is masked and reconstructed using Transformer architectures such as ViT [15]. Training objectives include patch reconstruction loss [3, 13] and discrete label prediction [8]. However, MAE representations often lack semantic alignment (*i.e.*, alignment of representations to capture intra-class similarities) as the reconstruction loss predominantly focuses on low-level time-frequency features while overlooking high-level semantic features [16, 17, 18]. As a result, they require significant amounts of labeled data for effective adaption to downstream tasks. To address this limitation, BEATs [8] trains an acoustic tokenizer alongside an audio self-supervised model iteratively, albeit at the cost of increased model complexity and training time, yielding only marginal improvements in downstream tasks and performing less optimally in low/few-shot scenarios.

In contrast, ID methods, such as contrastive learning (CL), semantically align representations of different augmentations of the same audio input [10]. Specifically, CL brings multiple augmentations of the same example (positive samples) closer while pushing other examples (negative samples) farther apart by utilizing an instance classification pretext strategy [2]. In the image domain, Lehner et al. [19] proposes to combine CL with Masked Image Modeling (MIM) to extract object-centric representations by disregarding background details, thereby alleviating substantial annotation requirements in downstream tasks. However, this approach requires large-scale unlabeled datasets [20], resulting in increased training time and computational overhead. Thus, combining ID and MAE to tackle downstream tasks with constrained labeled data remains challenging.

Our contributions. In this work, we introduce *uaMix-MAE*, an efficient ID contrastive tuning strategy with unsupervised audio Mixtures for pretrained MAEs, which enables effective adaptation to downstream tasks, particularly in low/few-shot settings, while only requiring small amounts of unlabeled data for MAE model tuning. uaMix-MAE initializes a ViT encoder with model weights trained with MAM [7] and tunes the model, using a contrastive objective, with unsupervised audio mixtures to semantically align representations of pretrained MAEs. Moreover, inspired by [21, 22], we propose a mixture technique tailored for audio that manipulates both the input and virtual label spaces simultaneously. This en-

courages the model to learn more precise and smoother decision boundaries in the latent feature space while training with small amounts of unlabeled data. Experimental results on several benchmark datasets show that uaMix-MAE outperforms strong MAM baselines by $4-6\%$ in low/few-shot scenarios.

2. RELATED WORK

Masked Audio Modeling (MAM) has been applied to various audio understanding [7, 8, 23, 24], natural language processing [6], and computer vision tasks [15]. As masked audio models, such as AudioMAE [3], MaskSpec [23], MSM-MAE [24], BEATs [8], and M2D [25], learn low-level features by reconstructing individual masked patches during training, they incorporate irrelevant background information and are prone to semantic misalignment. Thus, they perform poorly in downstream tasks with limited labeled data, such as few-shot learning. This work investigates the integration of ID and MAEs for improving adaptation to downstream tasks.

Instance Discrimination (ID) methods, unlike MAM, align representations of different augmentations of an anchor example. Existing works utilize data augmentation techniques such as pitch/time shift [14], time mask/stretch [14], random crop and mixup [10], fade [14], mixed/white noise [14], and Gaussian noise [10, 11]. However, none of these methods apply ID for semantically aligning representations of pretrained MAEs. To the best of our knowledge, we are the first to propose an efficient CL strategy with unsupervised audio mixtures to semantically align pretrained MAE representations using only a small amount of unlabeled data. Our work is akin to recent work in the image domain [22] aiming to reduce unlabeled data requirements, and consequently, computational resources and training time by training Transformers with ID [19].

3. METHODOLOGY

Given a pretrained MAE encoder f_θ and an unlabeled dataset $\mathcal{E} = \{(e_i, e_i^+)\}_{i=1}^N$, where N is the number of total examples, $e_i \in \mathbb{R}^{T \times F}$ denotes the Filterbanks (fbanks) [26] of an audio sample i with time T and frequency F, and e_i^+ is a positive example of anchor e_i, constructed through data augmentation techniques, our objective is to improve downstream task performance with limited labeled data. To achieve this, we can employ ID methods to leverage unlabeled data for semantically aligning the representations of f_θ in the feature space. In practice, however, training with abundant unlabeled data is impractical for resource-constrained environments. Therefore, devising methods that extract maximal value from unlabeled data can augment the transferability and generalization capabilities of the learned representations while training with a small amount of unlabeled data. To this end, we introduce uaMix-MAE, which extends f_θ by incorporating a contrastive head h_θ and performs contrastive tuning on h_θ and
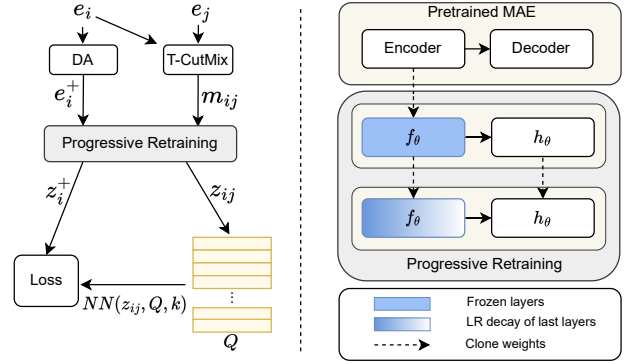

Fig. 1. uaMix-MAE overview. Left: T-CutMix contrastive tuning. Right: Progressive retraining of f_θ and h_θ. DA: Data Augmentation.

the last layers of f_θ. As the last layers capture more abstract and high-level features, contrastive tuning thus enhances the model's ability to utilize high-level semantics for downstream tasks. Fig. 1 presents the overall architecture of uaMix-MAE.

Contrastive Tuning Objective. In terms of ID, we utilize the Nearest Neighbour Contrastive Learning (NNCLR) objective [27], an extension of SimCLR [2] that utilizes a queue Q for the nearest neighbor lookup of anchor examples. Specifically, given a batch $B = \{(z_i, z_i^+)\}_{i=1}^{|B|}$, where z_i and z_i^+ are the feature representations of anchor e_i and its positive example e_i^+, respectively, and z_j denotes an example in B, the NNCLR loss function $\mathcal{L}_{CL}(z_i, z_i^+)$ is defined as follows:

$$\mathcal{L}_{CL}(z_i, z_i^+) = -\log \frac{\exp\left(NN(z_i, Q, k) \cdot z_i^+/\tau\right)}{\sum_{(z_j, z_j^+) \in B} \exp\left(NN(z_i, Q, k) \cdot z_j^+/\tau\right)} \quad (1)$$

Here, τ is a temperature hyperparameter and $NN(z_i, Q, k)$ denotes the top-k nearest neighbors of z_i. The queue Q is maintained similarly to MoCo [1]. The use of NNCLR as the contrastive tuning objective is motivated by its ability to provide more semantic variations in the positive examples compared to other methods [1, 2].

Progressive Retraining. For contrastive tuning, we employ a progressive retraining strategy. Initially, we freeze the pretrained MAE encoder f_θ and train the head h_θ using the NNCLR objective. We then retrain the second half of f_θ along with the head h_θ using NNCLR. Partial retraining of f_θ is motivated by the success of partial finetuning, *i.e.*, tuning only the last layers [3]. Intuitively, the lower layers of the encoder are adept at generalization, capturing fundamental features that apply across various contexts, while retraining the upper layers enables high-level semantic alignment of features and invariance to subtle differences among examples.

Unsupervised T-Cutmix. Traditional audio augmentation methods typically construct a positive example e_i^+ by manipulating the audio sample e_i solely in the input
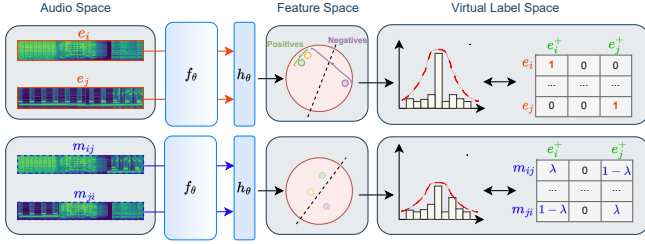
Fig. 2. (Top row) When original audio samples e_i, e_j are passed through f_θ and h_θ, the positive pair is close to each other, and the negative pair lies far in the feature space, resulting in a sharp decision boundary in the virtual label space. (Bottom row) uaMix-MAE creates mixed audio samples m_{ij} and m_{ji} in the input space and uses a softened distance function in the virtual label space, resulting in a smoother decision boundary.

space. These techniques include pitch/time shift [14], time mask/stretch [14], noise [10, 11, 14], random crop and mixup [10], *etc.* However, as depicted in Fig. 2, manipulating examples only in the input space results in a sharp decision boundary and, consequently, necessitating a large amount of data to learn generalized representations in the feature space [28, 29]. In contrast, we introduce T-Cutmix, an unsupervised mixing technique tailored for audio that manipulates data in both the audio and virtual label space as illustrated in Fig. 2. Our approach is akin to recent advances in image-based unsupervised mixing strategies such as MixUp [28] and 2D-CutMix [29], which combine label smoothing and self-supervised virtual label space regularization [30, 31, 15, 22]. Specifically, we define v_i as the virtual label of e_i and e_i^+, where $v_i[i] = 1$ signifies that e_i^+ is the positive example of e_i and $v_i[k \neq i] = 0$ indicates that all other audio samples in the batch B are considered negative examples in relation to e_i (Fig. 2 Virtual Label Space).

uaMix-MAE creates an audio mixture m_{ij} and its corresponding smoothed label y_{ij} as follows:

$$m_{ij} = \mathcal{M} \odot e_i + (1 - \mathcal{M}) \odot e_j \tag{2}$$

$$y_{ij} = \lambda v_i + (1 - \lambda) v_j, \tag{3}$$

where λ is a mixing coefficient and $\mathcal{M} \in \{0, 1\}^{T \times F}$ is a binary mask that determines which regions of an audio sample e_i are replaced with corresponding regions from e_j, *i.e.*, how much information from each sample contributes to the mixture (Audio Space in Fig. 2). To generate $\mathcal{M}$, a bounding box $BB = (s_t, 0, w_t, F)$ is sampled, indicating that BB in e_i is replaced with the patch cropped from BB of e_j. Here, $w_t = T\sqrt{1 - \lambda}$ denoting the length of BB in the time dimension. The starting time coordinate s_t is uniformly sampled as $s_t \sim \text{Uniform}(0, T)$. For the starting coordinate and the length in the frequency dimension of BB, we keep the values fixed at 0 and F, respectively. Similar to [30, 31], λ is sampled from a beta distribution $\mathcal{B}(\alpha, \alpha)$, where α is a hyperparameter controlling the size of BB. $\mathcal{M}$ is computed by filling the bounding box region with 0 and the rest with 1.

Table 1. Dataset details for each evaluation setting. SSL: Self-supervised training, FT: Finetuning, FS: Few-shot learning

Dataset	Purpose	# Classes	# Samples	Audio Length
AudioSet-20K [32]	SSL & FT	527	20,550	10s
ESC-50 [33]	FT & FS	50	2,000	5s
VoxCeleb1 [34]	FT & FS	1,251	153,516	3s - 180s
SCv2 [35]	FT & FS	35	105,829	1s
NSynth [36]	FS	1,006	305,978	4s
Kaggle18 [37]	FS	41	11,073	0.3s - 30s

Finally, the loss function for audio mixture m_{ij} is defined as

$$\mathcal{L}(z_{ij}, y_{ij}) = -\sum_{l=1}^{|B|} y_{ij}[l] \cdot \mathcal{L}_{CL}(z_{ij}, z_l^+), \tag{4}$$

where z_{ij} is the representation of the mixed example m_{ij}, $y_{ij}[l]$ the l-th element of the smoothed virtual label y_{ij}, and z_l^+ the representation of the positive example in the batch.

4. EXPERIMENTS

We evaluate uaMix-MAE on few-shot learning (FS) and fine-tuning (FT) downstream tasks across six benchmarks.

Baselines and Datasets. We compare uaMix-MAE with other self-supervised MAEs trained with MAM. For all baselines, we use the publicly available pretrained checkpoints. The datasets used in the self-supervised training and the downstream tasks are detailed in Table 1. For finetuning (FT), we use the AudioMAE train/validation/test splits [7].

Training details. The backbone f_θ is a 86M-parameter ViT-Base architecture. First, we initialize f_θ with pretrained AudioMAE encoder weights [7] and train h_θ for 40 epochs with learning rate 10^{-4}, batch size 512, temperature $\tau = 0.15$, and $k = 1$ in top k-NN lookup. Next, we freeze the lower half layers and train the top half layers by applying a layer-wise learning rate decay with decay factor 0.65, learning rate 10^{-4} for 160 epochs, and batch size 128. Other hyperparameters are adopted from the NNCLR initialization and contrastive tuning steps in [19]. For few-shot learning, we follow [38], and employ a nearest-centroid classifier on backbone extracted features. We report average accuracy (95% confidence interval) on 600 randomly sampled few-shot episodes. For fine-tuning experiments, we follow the AudioMAE setup [7].

4.1. Few-shot Learning

Table 2 presents the 5-way 1-shot (5 classes, each with 1 example) comparison of uaMix-MAE against baselines using prototypical networks. uaMix-MAE outperforms the best baseline by 4.90%–7.44% in all datasets except SCv2.

4.2. Few-shot Ablation Studies

Varying N and K. We vary N and K in the N-way K-shot experiments, where N is the class number, and K the number

Table 2. 5-way 1-shot performance using prototypical networks. Best performance is in **bold**. #TP refers to # of trainable parameters.

Method	#TP	ESC-50	VoxCeleb1	NSynth	SCv2	Kaggle18
MAE-AST [9]	99M	49.3±0.9	25.6±0.5	48.7±0.9	26.6±0.5	38.4±0.8
MaskSpec [23]	86M	43.0±0.7	-	-	21.1±0.4	-
BEATs [8]	90M	48.6±0.8	25.9±0.5	68.8±0.9	26.9±0.5	35.0±0.8
M2D [25]	86M	53.3±0.9	28.4±0.6	43.8±0.9	30.1±0.5	37.8±0.8
AudioMAE [7]	86M	61.1±0.9	28.9±0.5	70.6±0.9	**30.4±0.5**	41.3±0.8
uaMix-MAE	50M	**66.3±0.9**	**30.3±0.5**	**75.9±0.9**	29.6±0.5	**43.6±0.8**

Table 3. 5-way 1-shot performance comparison among uaMix-MAE variants: No Mixing, MixUp + LS. Best performance in **bold**.

Method	ESC-50	VoxCeleb1	NSynth	SCv2	Kaggle18
No Mixing	48.7±0.8	23.8±0.4	73.2±0.9	**29.9±0.5**	37.2±0.8
MixUp + LS	62.6±0.9	29.1±0.5	73.9±0.9	29.6±0.5	41.8±0.8
uaMix-MAE	**66.3±0.9**	**30.3±0.5**	**75.9±0.9**	29.6±0.5	**43.6±1.8**

(a) **N-way 1-shot**   (b) **5-way k-shot**

Fig. 3. N-way k-shot performance comparison on VoxCeleb1.

(a) **N-way 1-shot**   (b) **5-way k-shot**
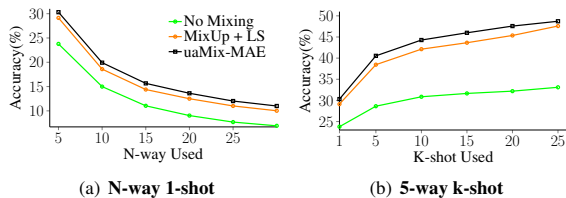
Fig. 4. Few-shot performance comparison on Voxceleb1 among uaMix-MAE variants: No Mixing and MixUp + LS.

of examples per class. Fig. 3 shows uaMix-MAE consistently outperforms baselines across different values of N and K.

T-CutMix Importance. We perform an ablation study considering the following variations: 1) No Mixing *i.e.*, employing no unsupervised mixing in CL, and 2) MixUp + LS *i.e.*, utilizing MixUp with label smoothing for unsupervised mixing. Results in Table 3 and Fig. 4 show substantial improvements compared to both variants across all scenarios.

TF-CutMix. To illustrate the impact of applying CutMix exclusively in the time dimension, we conduct an ablation study introducing a variation, termed uaMix-MAE-TF-CutMix, that employs CutMix in both time (T) and frequency (F) dimensions. As depicted in Fig. 5, uaMix-MAE with T-CutMix consistently outperforms uaMix-MAE with TF-CutMix.

4.3. Fine-tuning

Table 4 presents a fine-tuning comparison with uaMix-MAE demonstrating comparable results w.r.t. other baselines across

Table 4. Fine-tuning performance on audio and speech classification tasks. Best performance in **bold**.

Method	#TP	AudioSet-20k	ESC-50	VoxCeleb1	SCv2
BEATs [8]	90M	36.0	94.0	-	**98.3**
MAE-AST [9]	99M	30.6	90.0	63.3	97.9
MaskedSpec [23]	86M	32.3	89.6	-	97.7
AudioMAE [7]	86M	36.7	94.0	93.5	97.9
uaMix-MAE	86M	**37.0**	**94.1**	**93.6**	98.0

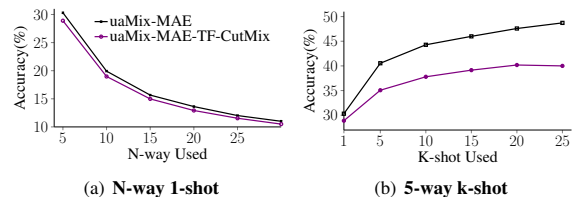

(a) **N-way 1-shot**   (b) **5-way k-shot**

Fig. 5. Few-shot performance comparison between uaMix-MAE and uaMix-MAE-TF-CutMix on VoxCeleb1.
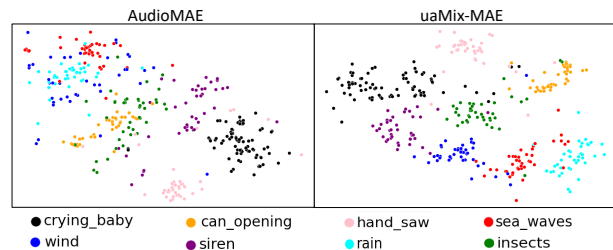

Fig. 6. t-SNE visualization of AudioMAE (left) and uaMix-MAE (right) features for eight ESC-50 classes.

all datasets. Table 2 and 4 indicate that uaMix-MAE achieves better performance in few-shot learning, *i.e.*, demonstrating superior generalization in the feature space while maintaining competitive results in fine-tuning.

4.4. Qualitative Analysis

We compare the learned feature representations of the AudioMAE encoder and uaMix-MAE. The t-SNE visualization [39] for eight ESC-50 classes (Fig. 6) reveals that uaMix-MAE exhibits better intra-class clustering compared to AudioMAE. Specifically, uaMix-MAE representations form distinct and well-separated clusters for classes 'sea_waves', 'wind', 'siren', and 'rain' while the AudioMAE representations for these classes overlap with other classes.

5. CONCLUSION

In this paper, we introduce uaMix-MAE, a contrastive tuning strategy employing unsupervised audio mixtures. To adapt to downstream tasks with limited labeled data, uaMix-MAE tunes a pretrained MAE encoder with a small amount of unlabeled data by mixing examples in both input and virtual label spaces. Experiments in few-shot settings demonstrate that uaMix-MAE outperforms existing masked audio models.

References

[1] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," In *CVPR*, 2020. 1, 2

[2] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," In *ICML*, 2020. 1, 2

[3] K. He, X. Chen *et al.*, "Masked autoencoders are scalable vision learners," In *CVPR*, 2022. 1, 2

[4] T. Brown, B. Mann, N. Ryder *et al.*, "Language models are few-shot learners," In *NeurIPS*, 2020. 1

[5] H. Touvron, T. Lavril, G. Izacard *et al.*, "Llama: Open and efficient foundation language models," *arXiv:2302.13971*, 2023. 1

[6] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," In *NAACL-HLT*, 2019. 1, 2

[7] P. Huang, H. Xu, J. Li *et al.*, "Masked autoencoders that listen," In *NeurIPS*, 2022. 1, 2, 3, 4

[8] S. Chen, Y. Wu, C. Wang *et al.*, "BEATs: Audio pre-training with acoustic tokenizers," In *ICML*, 2023. 1, 2, 4

[9] A. Baade, P. Peng, and D. Harwath, "MAE-AST: Masked autoencoding audio spectrogram transformer," In *Interspeech*, 2022. 1, 4

[10] D. Niizumi, D. Takeuchi, Y. Ohishi *et al.*, "Byol for audio: Self-supervised learning for general-purpose audio representation," In *IJCNN*, 2021. 1, 2, 3

[11] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," In *ICASSP*, 2021. 1, 2, 3

[12] H. Wu, P. Seetharaman *et al.*, "Wav2clip: Learning robust audio representations from clip," In *ICASSP*, 2022. 1

[13] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," In *Interspeech*, 2021. 1

[14] C. Heggan, T. Hospedales, S. Budgett, and M. Yaghoobi, "MT-SLVR: Multi-Task Self-Supervised Learning for Transformation In(Variant) Representations," In *Interspeech*, 2023. 1, 2, 3

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," In *ICLR*, 2020. 1, 2, 3

[16] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," In *ICLR*, 2021. 1

[17] A. Ramesh, M. Pavlov, G. Goh *et al.*, "Zero-shot text-to-image generation," In *ICML*, 2021. 1

[18] B. Epstein and R. Meir, "Generalization bounds for unsupervised and semi-supervised learning with autoencoders," *arXiv:1902.01449*, 2019. 1

[19] J. Lehner, B. Alkin, A. Fürst *et al.*, "Contrastive tuning: A little help to make masked autoencoders forget," *arXiv:2304.10520*, 2023. 1, 2, 3

[20] O. Russakovsky, J. Deng, H. Su *et al.*, "Imagenet large scale visual recognition challenge," *IJCV*, 2015. 1

[21] Z. Shen, Z. Liu, Z. Liu *et al.*, "Un-mix: Rethinking image mixtures for unsupervised visual representation learning," In *AAAI*, 2022. 1

[22] H. Y. Y. Cao and J. Wu, "Training vision transformers with only 2040 images," In *ECCV*, 2022. 1, 2, 3

[23] D. Chong, H. Wang, P. Zhou, and Q. Zeng, "Masked spectrogram prediction for self-supervised audio pre-training," In *ICASSP*, 2023. 2, 4

[24] D. Niizumi, D. Takeuchi, Y. Ohishi *et al.*, "Masked spectrogram modeling using masked autoencoders for learning general-purpose audio representation," In *HEAR*, 2022. 2

[25] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Masked Modeling Duo: Learning Representations by Encouraging Both Networks to Model the Input," In *ICASSP*, 2023. 2, 4

[26] https://pytorch.org/audio/main/generated/torchaudio.compliance.kaldi.fbank.html. 2

[27] D. Dwibedi, Y. Aytar, J. Tompson *et al.*, "With a little help from my friends: Nearest-neighbor contrastive learning of visual representations," In *ICCV*, 2021. 2

[28] H. Zhang, M. Cisse, Y. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," In *ICLR*, 2018. 3

[29] S. Yun, D. Han, S. Oh *et al.*, "Cutmix: Regularization strategy to train strong classifiers with localizable features," In *ICCV*, 2019. 3

[30] K. Lee, Y. Zhu *et al.*, "i-Mix: A domain-agnostic strategy for contrastive representation learning," In *ICLR*, 2021. 3

[31] Z. Shen, Z. Liu *et al.*, "Un-mix: Rethinking image mixtures for unsupervised visual representation learning," In *AAAI*, 2022. 3

[32] J. Gemmeke, D. Ellis *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," In *ICASSP*, 2017. 3

[33] K. Piczak, "ESC: Dataset for environmental sound classification," In *ACM MM*, 2015. 3

[34] A. Nagrani, J. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," In *Interspeech*, 2017. 3

[35] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv:1804.03209*, 2018. 3

[36] J. Engel, C. Resnick *et al.*, "Neural audio synthesis of musical notes with wavenet autoencoders," In *ICML*, 2017. 3

[37] E. Fonseca, M. Plakal *et al.*, "General-purpose tagging of freesound audio with audioset labels: Task description, dataset, and baseline," In *DCASE*, 2018. 3

[38] L. Ericsson, H. Gouk, and T. Hospedales, "How well do self-supervised models transfer?" In *CVPR*, 2021. 3

[39] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *JMLR*, 2008. 4