CLAR: Contrastive Learning of Auditory Representations

Haider Al-Tahan
haltaha@uwo.ca
Department of Computer Science
The University of Western Ontario
London, Ontario
Canada

Yalda Mohsenzadeh
ymohsenz@uwo.ca
Department of Computer Science
The University of Western Ontario
London, Ontario
Canada

Abstract

Learning rich visual representations using contrastive self-supervised learning has been extremely successful. However, it is still a major question whether we could use a similar approach to learn superior auditory representations. In this paper, we expand on prior work (SimCLR) to learn better auditory representations. We (1) introduce various data augmentations suitable for auditory data and evaluate their impact on predictive performance, (2) show that training with time-frequency audio features substantially improves the quality of the learned representations compared to raw signals, and (3) demonstrate that training with both supervised and contrastive losses simultaneously improves the learned representations compared to self-supervised pre-training followed by supervised fine-tuning. We illustrate that by combining all these methods and with substantially less labeled data, our framework (CLAR) achieves significant improvement on prediction performance compared to supervised approach. Moreover, compared to self-supervised approach, our framework converges faster with significantly better representations [1].

[1]Code is available at: `https://github.com/haideraltahan/CLAR`

Proceedings of the 24$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

1 INTRODUCTION

Although humans are proficient at perceiving and understanding sounds, making algorithms perform the same task poses a challenge due to the wide range of variations in auditory features. Applications of sound understanding range from surveillance (Radhakrishnan et al., 2005) and music classification (Choi et al., 2017; Ibrahim et al., 2020) to audio generation (Engel et al., 2019; Donahue et al., 2019) and deep-fake detection (Mittal et al., 2020).

Achieving automated auditory perception requires the learning of effective representations. Often prior work derive effective representations through discriminative approaches (Park et al., 2019a; Hershey et al., 2017; Tokozume and Harada, 2017; Guzhov et al., 2020). That is, similar to supervised learning, the model learns the mapping between the input signal to the class label. The underlying assumption with such approach is that the latent representations carry effective representations for the designed tasks. One fundamental problem with such learned representations is the potential limitation to generalizability. First, those representations are only limited to availability of expensive and time consuming labeled data. Secondly, representations are skewed towards one particular domain (e.g. speech, music, etc ...). Therefore, in both cases, major fine-tuning to the targeted training data would be required. Alternatively, recent self-supervised approaches using contrastive learning in the latent space have been shown to learn efficient representations that achieves state-of-the-art performance in images (Chen et al., 2020; Qian et al., 2020; Bachman et al., 2019; Oord et al., 2018; Dosovitskiy et al., 2014; Hadsell et al., 2006) and videos (Qian et al., 2020). However, it is still a major question on how we can achieve similar landmark on auditory data.

In this work, we build on SimCLR (Chen et al., 2020), a self-supervised framework for contrastive learning of visual representations. We show that similar framework could be adopted for learning effective auditory representations. Moreover, with a simple modification, we are able to reduce the training time and improve recognition performance.

In order to accomplish this, we introduce four major components that are important to nourish the learning of auditory representations. We:

- Demonstrate the success of contrastive learning in learning efficient auditory representations.
- Investigate six data augmentation operations and show their effect on auditory classification task both with raw audio and extracted time-frequency audio features.
- Show that training with time-frequency audio features substantially improves the quality of the learned representations in contrastive learning compared to raw audio signals.
- Show that using supervised and contrastive learning simultaneously while training, not only improves the learned representations but also speeds up the training.

By combining all these methods, our framework (CLAR) achieves 96.1% top-1 accuracy, which is 1% relative improvement over supervised method and 11.3% improvement over SimCLR on Speech Command dataset. Moreover, compared to SimCLR, our framework converges faster with significantly better representations.

2 RELATED WORK

2.1 Contrastive Learning

In an era of ever increasing unlabeled data, contrastive learning has been shown to be effective at capitalizing on such data (Hadsell et al., 2006; Dosovitskiy et al., 2014; Wu et al., 2018; Zhuang et al., 2019; Tian et al., 2020a,b). Contrastive learning is a self-supervised framework that formulates representations in a given model based on similarity/dissimilarity of a given input pairs. Recently proposed method called SimCLR has been shown to not only outperform previous self-supervised methods on ImageNet but also outperform supervised methods on some natural image classification datasets (Chen et al., 2020). In essence, the framework aims at learning efficient representations by maximizing agreement between differently augmented views of the same data and maximizing difference across contrasting images via contrastive loss in the latent space. The loss for such objective is termed Normalized Temperature-scaled Cross Entropy Loss (NT-Xent):

$$\mathcal{L}_{CL} = -\sum_{i,j}^{N} \log \frac{\exp\left(\operatorname{sim}\left(\mathbf{z}_i, \mathbf{z}_j\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbf{1}_{[k\neq i]} \exp\left(\operatorname{sim}\left(\mathbf{z}_i, \mathbf{z}_k\right)/\tau\right)} \tag{1}$$

where $\mathbf{1}_{[k\neq i]} \in 0,1$ is an indicator function evaluating to 1 iff $k \neq i$, τ denotes a temperature parameter (default is 0.5), and $\mathbf{z}$ denotes the encoded representations of a given augmented view. N is the number of training samples within a mini-batch, (i, j) are positive pairs of each sample. The loss is computed across all positive pairs, in a mini-batch. $\operatorname{sim}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T\mathbf{v}/||\mathbf{u}||||\mathbf{v}||$ denote the cosine similarity between two vectors $\mathbf{u}$ and $\mathbf{v}$.

Our work extends SimCLR by introducing an approach to learning auditory representations instead of visual representations. We do that by showing composition of augmentations that are most efficient in learning auditory representations. Previous work on auditory data augmentations have shown effective methods that improve supervised classifications applied to both on raw audio signal (McFee et al., 2015; Schlüter and Grill, 2015; Ko et al., 2015) and time-frequency audio features (Park et al., 2019b). However, auditory data augmentations that nourish effective representations with contrastive learning is yet to be investigated. In this work, we also investigate the effect of the augmentations on learning representations from both raw audio signal and time-frequency audio features in contrastive learning framework.

2.2 Supervised Contrastive Learning

A common practice in deep learning is to train parameters of a model in a supervised manner using cross-entropy loss (Rumelhart et al., 1986; Baum and Wilczek, 1988).

$$\mathcal{L}_{CE} = -\sum_{i}^{C} t_{i,o} \log(p_o) \tag{2}$$

where C is the number of classes, t_i is a binary indicator (i.e. 0 or 1) if observed class label o is the same as the class label i and p is the predicted probability of the observation o. Several studies has identified considerable drawbacks in utilizing cross-entropy loss, such as sensitivity to hyper-parameters (Khosla et al., 2020), sensitivity to noisy labels (Sukhbaatar et al., 2015), and poor margins (Cao et al., 2019). Khosla et al. (2020) demonstrated a promising approach to boost some of the drawbacks of cross-entropy loss by modifying contrastive loss to leverage labeled information in learning efficient visual representation (SupCon). In short, self-supervised contrastive loss (e.g. NT-Xent) contrasts augmented versions of the same image with the reminder of the batch. While, SupCon contrasts the set of all samples from the same class against the remainder of the batch. Thereby making such approach strictly supervised and limits the method to labeled data. Alternatively, our approach incorporate both self-supervised frameworks and supervised training (rather than self-supervised pre-training and supervised fine-tuning). Our method not only could foster more efficient representations but also speed the training process. Furthermore, unlike Khosla et al. (2020), our proposed method can leverage unlabeled and labeled data (even if limited labeled data is available) simultaneously during training.

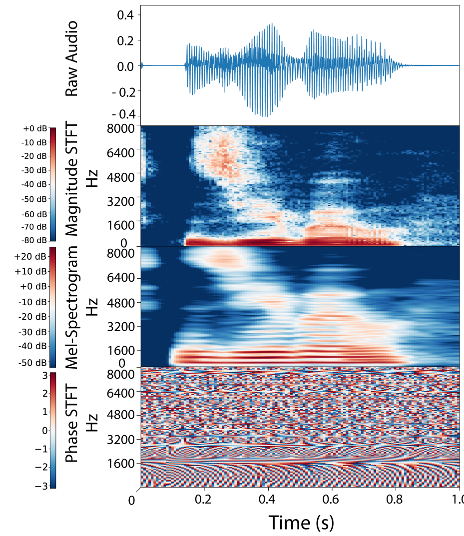


Figure 1: Raw audio (first) with the subsequent time-frequency features, specifically, STFT (short-time Fourier transform) magnitudes (second) and phase angles (fourth), and Mel-spectorgram (third). The raw audio and time-frequency features were utilized in training 1D and 2D versions of ResNet.

3 METHODS

3.1 Audio Pre-porcessing

We trained two family of models, one that takes as input raw audio signals while the other utilizes time-frequency audio features as input. For both models, we start by down-sampling (when applicable) all audio signals to 16kHz, followed by signal padding (by zeros) or clipping the right side of the signal to ensure that all audio signals are of the same length. The target length of the audio signal is set based on the datasets' assigned audio length (Section 3.3). For models dependant on time-frequency features, we computed the short-time Fourier transform (STFT) magnitudes and phase angles of the input audio with 16 ms windows and 8 ms stride (Allen, 1977), further we projected the STFT to 128 frequency bins equally spaced on the Mel scale (Figure 1). Moreover, we computed the log-power of magnitude STFT and Mel spectrogram (Eq. 3) and stacked the three time-frequency features in the channel dimension, resulting in a matrix of size: $3 \times F \times T$. Where F is the number of the frequency bins and T is the number of frames in the spectrogram. This was done to ensure that we have comprehensive features that could capture multi-domain audio signals (e.g. speech, environmental and music sounds) and would not require us to change the baseline ResNet 2D architecture. All spectrogram transformations were performed on GPU using a 1D Convolutional Neural Network (Cheuk et al., 2020).

$$f(S) = 10log_{10}|S|^2 \tag{3}$$

where S is the mel-spectrogram or magnitude STFT.

3.2 Training/Evaluation Protocol

To maintain consistency and allow for a fair comparison across supervised, self-supervised and our proposed method (CLAR), we adopted four components in our framework (see Figure 2):

1. **Data Augmentation** module generates two random views of each sample, which contains an augmented information of the original sample. Section 5 describes all the various augmentations that were used in our experiments and Section 6 provides the effect of these augmentations on auditory classification task both with raw audio and extracted time-frequency audio features.

2. **Encoder** maps the data samples into a representational vector. To vectorize our representations we performed adaptive average pooling on the output of the encoder. For our encoder, we trained 1D and 2D variants of standard ResNet18 (He et al., 2016) with SimCLR training protocol. For models trained on time-frequency audio features (i.e. spectrograms) as input, we utilized a typical ResNet18 (Hershey et al., 2017) with random initialization. Alternatively, we switched all ResNet18 operations such as convolutions, max-pooling and batch normalization from 2D to 1D for models that takes raw audio signal as input. The output of both models is a 512 dimension vector.

3. **Projection head** maps the extracted encoder representations to a space where contrastive/supervised loss is computed. Projection head consists of three fully connected layers with ReLU activation functions. We performed the losses on the output of the projection head with fixed vector size of 128. In the supervised approach we replaced the final linear layer used for contrastive loss with a linear layer with the size of the class numbers to compute cross entropy loss. Lastly, in our proposed approach, in addition to the cross entropy loss computed on the last layer, we computed contrastive loss using the representations in the layer preceding the last layer.

4. **Evaluation head** was used to replace the projection head after training the encoder using different training methods. The evaluation head is a linear classifier trained on top of the frozen encoder to assess the learned representation quality by computing the test accuracy. When limiting labeled data to compare performance across supervised, self-supervised and our proposed approach (Section 8), we trained the evaluation head on the full labeled data. This approach is commonly adopted to evaluate the learned representations of self-supervised methods (Chen et al., 2020; Kolesnikov et al., 2019; Bachman et al., 2019).

We trained all models with 1024 batch size, layer-wise adaptive rate Scaling (LARS) optimizer (You et al., 2017) with learning rate of 1.0, weight decay of 10^{-4}, linear warmup for the first 10 epochs, decay of the learning rate with the cosine decay schedule without restarts (Loshchilov and Hutter, 2016) and global batch normalization. For some datasets in section 7, we reduced the batch size to 512 to be able to fit the data in memory. In section 8 we trained all our models on augmentations that we found to yield the best performance on test accuracy (section 6 & 7). All models were trained from random initialization with 4 NVIDIA v100 Tesla 32GB GPUs.

3.3 Datasets

In this work, we evaluated the performance of our proposed framework on three audio datasets from different domains (speech, music, and environmental sounds). All datasets have predefined train-validation-test splits by authors (except the ECS dataset), we used the test splits to compute analysis. The datasets are:

- **Speech Commands** (Speech): composed of 105,829 16kHz single-channel audios (Warden, 2018) from 2,618 speakers. Each audio file contains a one second recording of a single spoken English word from limited vocabulary. The dataset contains 35 labels (words) such as one-digit numbers, action oriented words, and arbitrarily short words. In addition to the full dataset, we derive a simpler version ($\sim$20k samples) with only the utterances of the one-digit numbers.

- **NSynth** (Music): contains 305,979 four seconds audio of musical notes, each with a unique pitch and musical instrument family (Engel et al., 2017). For every musical note, the note was held for the first three seconds and allowed to decay for the final second. Similar to Speech Commands, we composed two variations of the same dataset, (1) we utilized musical instrument family as the class labels (11 classes) and (2) we used pitch as the class labels (128 classes). Both variations of NSynth dataset included the same amount of data, however, the number of classes varied.

- **Dataset for Environmental Sound Classification** (Environmental): consists of two variants ESC-10 and ESC-50 provided by the authors. Similar to previous datasets described, the two variants describe the number of classes. The ESC-50 dataset consists of 2,000 5-seconds environmental recordings equally distributed across 50 classes (40 clips per class). Classes such as animal, natural and water, non-speech human, interior and exterior sounds (Piczak, 2015). The ESC-10 is a subset of ESC-50 consisting of 400 recordings, making it the dataset with the least number of training data in our collection. Both datasets are divided into 5 folds by the authors. In this work, we utilized the first 4 folds for training and the last for testing.

4 CLAR Framework

Contrastive and supervised learning share the common goal of constructing representations that distinguish samples for different tasks. The supervised approach focuses on distinguishing samples from multiple classes without constraints on the latent representations. While, contrastive learning constructs such representations between paired views from samples with the constraint being that latent representations of negative views (from different samples) are maximized and positive views (from same samples) are minimized. These two frameworks have their own advantages and disadvantages. For instance, constrastive learning benefits from larger batch sizes and longer training (Goyal et al., 2018; Chen et al., 2020). However, the supervised approach is simpler to optimize,

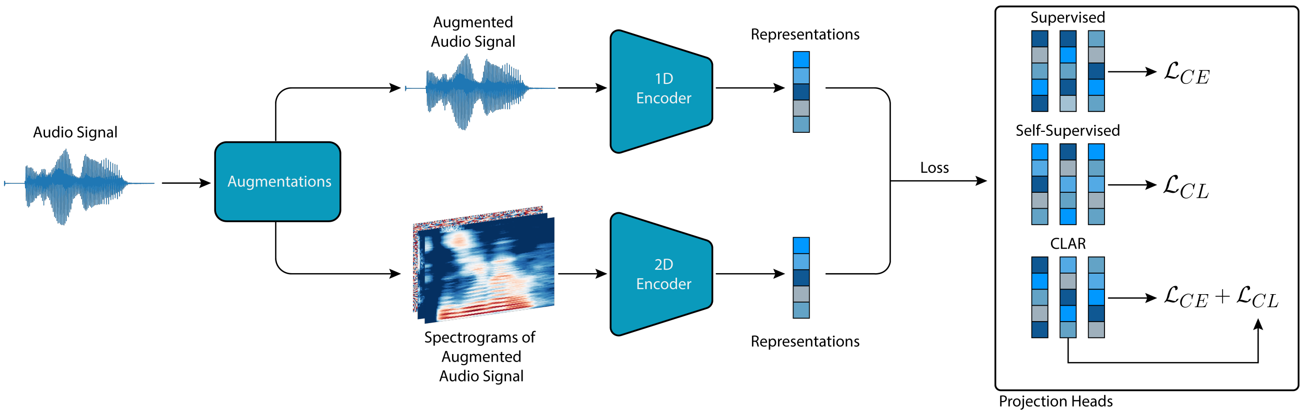


Figure 2: Training pipeline. We start by applying augmentations directly to the audio signal followed by either feeding the augmented audio signal or the spectrograms of the augmented audio signal to the subsequent encoder. We feed the representations from the encoder to the same projection head architecture across all approaches. However, we change the loss and the the layer representation employed in the loss computation.

hence, requires less training to achieve relative performance (Chen et al., 2020). Self-supervised contrastive learning aims to benefit from both frameworks by combining the shared representations from both frameworks by performing self-supervised pre-training followed by supervised fine-tuning on labeled examples (Chen et al., 2020; Hénaff et al., 2019; He et al., 2020; Kiros et al., 2015). However, this could result in problems such as *catastrophic forgetting*, especially in smaller networks (McCloskey and Cohen, 1989; Goodfellow et al., 2013; Li and Hoiem, 2017) and makes the training more difficult as there are two stages that would need to be optimized. In this work, we abolish the fine-tuning step and integrate both contrastive and supervised learning frameworks simultaneously during training:

$$L = \mathcal{L}_{CL} + \mathcal{L}_{CE} \tag{4}$$

where $\mathcal{L}_{CL}$ is the contrastive loss and $\mathcal{L}_{CE}$ is the Categorical Cross-Entropy (CE) loss of the labeled samples. In CE loss, in cases where the labels for some of the samples within the mini-batch are missing, then the CE loss will be set to zero. Alternatively, contrastive loss is always applied as it is not dependent on the labels. A possible approach to guarantee labeled samples within a mini-batch is to use stratified sampling, this is especially important when the labeled data is substantially small portion of the whole dataset (e.g. 1% of the data is labeled). In our analysis, we utilized random sampling because (1) datasets utilized are not large, especially when compared with ImageNet and (2) we utilized very large batch size (1024). Using the projection head, we apply the $\mathcal{L}_{CE}$ loss on the last layer of the projection head and the $\mathcal{L}_{CL}$ loss on the layer preceding the last layer of the projection head (Figure 2).

5 Augmentations

To investigate the impact of various data augmentations suitable for learning auditory representations, we deployed six distinct augmentations (Figure 3). Each augmentation was applied directly to the audio signal. Augmentations that directly influence spectrograms were not included (Park et al., 2019b) to ensure that we could make direct comparison of augmentations performance both with the 1D and 2D models. In each iteration during training, each augmentation has a hyper-parameter which is randomly sampled from a uniform distribution that result in either a degree of data transformation or none at all. Introduced augmentations could be categorised as either frequency or temporal transformations:

1. Frequency Transformations
   (a) **Pitch Shift (PS)**: randomly raises or lowers the pitch of the audio signal (McFee et al., 2020). Based on experimental observation, we found the range of pitch shifts that maintained the overall coherency of the input audio was in the range [-15, 15] semitones.
   (b) **Noise Injection**: mix the audio signal with random white, brown and pink noise. In our implementation, the intensity of the noise signal was randomly selected based on the strength of signal-to-noise ratio. We adopted two versions of this augmentation: (1) applied only white noise with varying degree of intensity (White Noise), (2) applied either

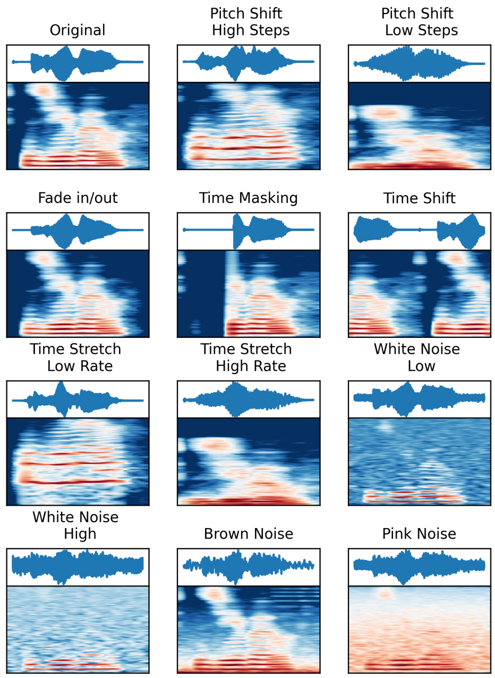


Figure 3: Raw audio with the subsequent mel-spectrogram for each audio augmentation utilized in the paper. Each augmentation have a varying degree of the shown transformation depending on random hyper parameters. Here an example of each augmentation is illustrated.

white, brown, or pink depending on an additional random parameter sampled from uniform distribution (Mixed Noise).

2. Temporal Transformations

   (a) **Fade in/out (FD)**: gradually increases/decreases the intensity of the audio in the beginning/end of the audio signal. The degree of the fade was either linear, logarithmic or exponential (applied with uniform probability of 1/3). The size of the fade for either side of the audio signal could at maximum reach half of the audio signal. The size of the fade was another random parameter picked for each sample.

   (b) **Time Masking (TM)**: given an audio signal, in this transformation we randomly select a small segment of the full signal and set the signal values in that segment to normal noise or a constant value. In our implementation, we not only randomly selected the location of the masked segment but also we randomly selected the size of the segment. The size of the masked segment was set to maximally be 1/8 of the input signal.

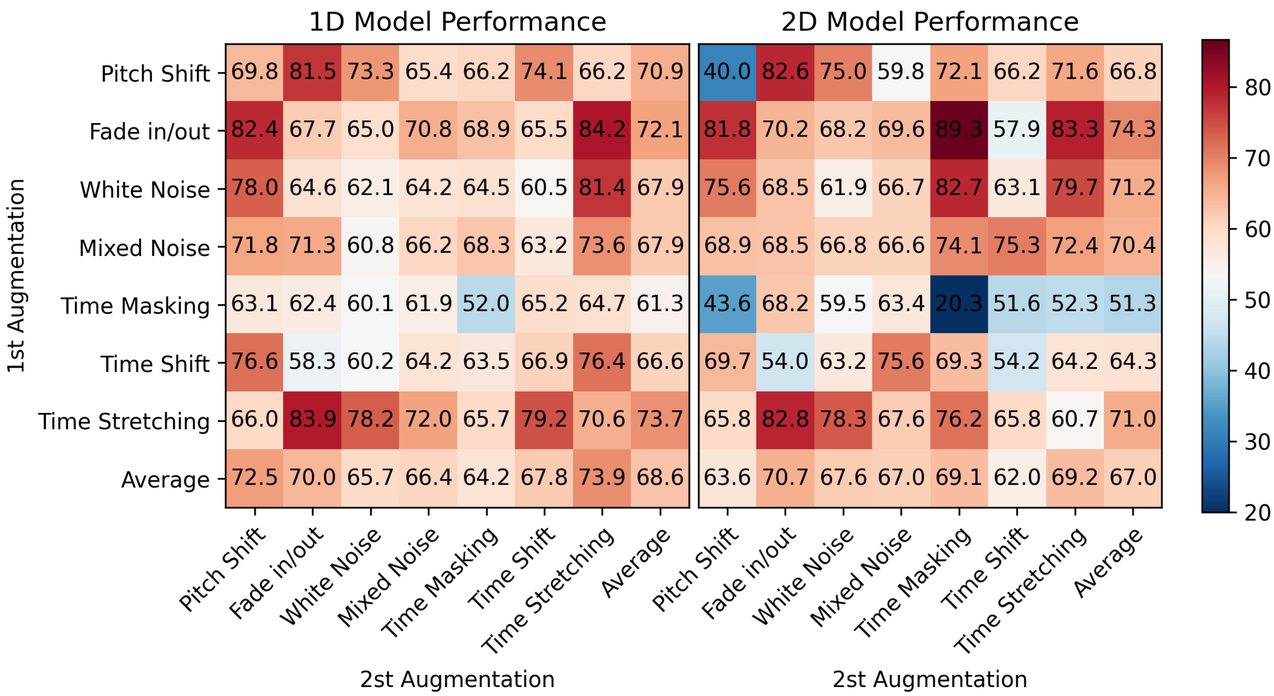
   (c) **Time Shift (TS)**: randomly shifts the audio samples forwards or backwards. Samples that roll beyond the last position are re-introduced at the first position (rollover). The degree and direction of the shifts were randomly selected for each audio. The maximum degree that could be shifted was half of the audio signal, while, the minimum was when no shift applied to the signal.

   (d) **Time Stretching (TST)**: slows down or speeds up the audio sample (while keeping the pitch unchanged). In this approach we transformed the signal by first computing the STFT of the signal, stretching it using a phase vocoder, and computing the inverse STFT to reconstruct the time domain signal (McFee et al., 2020). Following those transformations, we down-sampled or cropped the signal to match the same number of samples as the input signal. When the *rate* of stretching was greater than 1, the signal was sped up. Otherwise when the *rate* of stretching was less than 1, then the signal was slowed down. The *rate* of time stretching was randomized for each audio with range values of [0.5, 1.5].

6 DATA AUGMENTATIONS FOR CONTRASTIVE LEARNING

Data augmentations have been widely adopted in audio (McFee et al., 2015; Schlüter and Grill, 2015; Ko et al., 2015; Park et al., 2019b) and image (Krizhevsky et al., 2012; Hénaff et al., 2019) domains. Moreover, recently it has been shown that some sequences of augmentations in image domain can offer a relatively better performance compared to other augmentations in contrastive learning (Chen et al., 2020). In this section, we investigate the impact of different augmentations applied to the signal level on the quality of learned auditory representations.

To investigate the effect of individual auditory data augmentations and their sequential ordering, we perform comprehensive training of SimCLR framework on Speech Commands-10 dataset for 1000 epochs on all the proposed auditory augmentations (Section 5). Figure 4 shows top-1 test performance on both 1D and 2D variants of ResNet18. The diagonal line represents the performance of single augmentation, while other



1D: 1st \ 2nd	Pitch Shift	Fade in/out	White Noise	Mixed Noise	Time Masking	Time Shift	Time Stretching	Average
Pitch Shift	69.8	81.5	73.3	65.4	66.2	74.1	66.2	70.9
Fade in/out	82.4	67.7	65.0	70.8	68.9	65.5	84.2	72.1
White Noise	78.0	64.6	62.1	64.2	64.5	60.5	81.4	67.9
Mixed Noise	71.8	71.3	60.8	66.2	68.3	63.2	73.6	67.9
Time Masking	63.1	62.4	60.1	61.9	52.0	65.2	64.7	61.3
Time Shift	76.6	58.3	60.2	64.2	63.5	66.9	76.4	66.6
Time Stretching	66.0	83.9	78.2	72.0	65.7	79.2	70.6	73.7
Average	72.5	70.0	65.7	66.4	64.2	67.8	73.9	68.6

2D: 1st \ 2nd	Pitch Shift	Fade in/out	White Noise	Mixed Noise	Time Masking	Time Shift	Time Stretching	Average
Pitch Shift	40.0	82.6	75.0	59.8	72.1	66.2	71.6	66.8
Fade in/out	81.8	70.2	68.2	69.6	89.3	57.9	83.3	74.3
White Noise	75.6	68.5	61.9	66.7	82.7	63.1	79.7	71.2
Mixed Noise	68.9	68.5	66.8	66.6	74.1	75.3	72.4	70.4
Time Masking	43.6	68.2	59.5	63.4	20.3	51.6	52.3	51.3
Time Shift	69.7	54.0	63.2	75.6	69.3	54.2	64.2	64.3
Time Stretching	65.8	82.8	78.3	67.6	76.2	65.8	60.7	71.0
Average	63.6	70.7	67.6	67.0	69.1	62.0	69.2	67.0

Figure 4: Top-1 test performance on Speech Commands-10 for 1D (left matrix) and 2D (right matrix) models trained for 1000 epochs. The diagonal line represents the performance of single augmentation, while other entries represent the performance of paired augmentations. Each row shows the first augmentation and column shows the second augmentation applied sequentially.

entries represent the performance of paired augmentations. Each row indicates the first augmentation and each column shows the second augmentation applied sequentially. The last column and row in each matrix represents the averaged predictive performance of a specific augmentation. The last column depicts the average when the augmentation was applied first, while the last row shows the average when the corresponding augmentation was applied second. The bottom right element represents the average of the whole matrix. Similar to Chen et al. (2020), we found that multiple augmentations are required to learn efficient representations. In particular, with the 1D model, we observe that the composition of fade in/out, time stretching and pitch shifting significantly are the top-3 augmentations that improve the quality of representations. While on the 2D model, we observe that fade in/out, time masking and time shifting are premier in improving the quality of the auditory representations. On average the 1D variant of the model shows better performance (**1D**: 68.6±0.82; **2D**: 67.0±1.36), partially due time masking yielding the worse performance when applied as the first augmentation during the training of the 2D model. Moreover, the 2D variant of the model achieves the maximum recorded performance (89.3%).

7 RAW SIGNAL VERSUS TIME-FREQUENCY FEATURES

Time-frequency audio features have been used extensively in the literature, as algorithms trained on such features have consistently demonstrated better performance compared to algorithms trained on raw audio signals (Tokozume and Harada, 2017; Tokozume et al., 2017; Guzhov et al., 2020; Engel et al., 2017). In this section, we extend on this concept by investigating the efficiency of representations learned from raw signal and time-frequency features using contrastive learning.

Table 1 shows the accuracy of evaluation head attached to the frozen encoder trained with augmentations that yielded the best performance from section 6. We found that time-frequency features compared to raw audio signal consistently improve the learned representations. In particular, fade in/out and time masking using time-frequency features outperforms all other methods. We also found that increasing the number of augmentations does not necessary improve the learned representations. This was observed when predictive performance degraded after appending time-stretching augmentation to fade in/out and time masking.

Table 1: Accuracy of ResNet18 trained on various datasets using different augmentations.

Models	Datasets					
1D	ECS-10	ECS-50	SC-10	SC-50	NSynth-11	NSynth-128
FD	12.5	3.3	67.7	16.1	25.1	52.2
FD + TST	52.1	2.9	84.2	24.2	34.6	70.3
FD + TST + PS	10.0	3.5	79.9	23.6	46.2	10.8
2D						
FD	46.2	19.5	70.2	13.7	31.1	52.8
FD + TM	**68.7**	33.7	**89.3**	**29.1**	**48.8**	**84.1**
FD + TM + TS	62.5	**40.4**	84.6	25.0	48.6	80.4

Table 2: Accuracy of ResNet18 trained for 1000 empochs on Speech Command-10 dataset with incrementally less labels. During evaluation phase, we trained the evaluation head on all the labeled data with the frozen encoder.

Method	Type	Labeled Data Percentage			
		100%	20%	10%	1%
Cross-Entropy	Supervised	94.9	86.4	68.4	28.6
SupCon	Supervised	96.0	87.9	82.1	26.6
SimCLR	Self-supervised (Unsupervised)	84.8			
CLAR	Semi-Supervised	**96.1**	**90.5**	**89.1**	77.9

8 CLAR REPRESENTATIONS VERSUS SUPERVISED AND SELF-SUPERVISED

In this section, we utilize augmentations that yielded the best performance on SC-10 from section 7 to investigate the efficiency of CLAR with supervised and self-supervised methods. We utilize SC-10 as the dataset of choice and investigate both the predictive performance of the evaluation head (Section 3.2 for more explanation) every 10 epochs for a maximum of 1000 epochs. This would not only shed light on the final performance but also the speed at which the methods reach such performance. Moreover, as CLAR is capable of semi-supervised training, we test its capability by training models on 100%, 20%, 10% and 1% labeled data. For the self-supervised simCLR method we use no labeled data, hence, the performance would be the same across the board.

8.1 CLAR improves learned representations

Table 2 shows the top-1 accuracy of models trained using various methods while changing the percentage of labeled data. We found that the representations of the encoder trained with the CLAR method outperforms the representations learned using supervised and self-supervised methods when trained over the same number of epochs. When trained on 100% of the labeled data, CLAR achieves 96.1%, followed by SupCon (Khosla et al., 2020) with 96.0% and Cross-Entropy with 94.9%. Furthermore, we show that this trend continues as we decrease the labeled data. That is, while the supervised methods loses 65% of the performance as we decrease the labeled data from 100% to 1%, CLAR decreases by only 19%. These results show that CLAR indeed combines both supervised and self-supervised methods to draw more efficient representations. Lastly, we found that when we decrease the amount of labeled data to only 1%, the performance of CLAR's learned representation degrades compared to the self-supervised method. This could be the result of overfitting more to the labeled data resulting in less efficient representations.

8.2 CLAR improves the speed of learning representations

Figure 5 shows the top-1 test performance of SC-10 dataset computed every 10-epochs by training an evaluation head attached to a frozen encoder over 1000 epochs. We found that the CLAR method not only improves the representations in the encoder but also improves the speed at which those representation are learned compared to the self-supervised approach. In particular, when we provide 100% of the labeled data, CLAR shows not only better predictive performance compared to supervised and self-supervised method but also show better training speed than self-supervised. As we decrease the amount of labeled data, this trend continues while the gap between the supervised method and CLAR test performance increases. These results suggest that both the self-supervised and supervised training tasks indeed share common representations that improves latent representations in the encoder. Moreover, the training algorithm optimize for the $\mathcal{L}_{CE}$ loss first followed by gradual slow optimization of the $\mathcal{L}_{CL}$ loss.

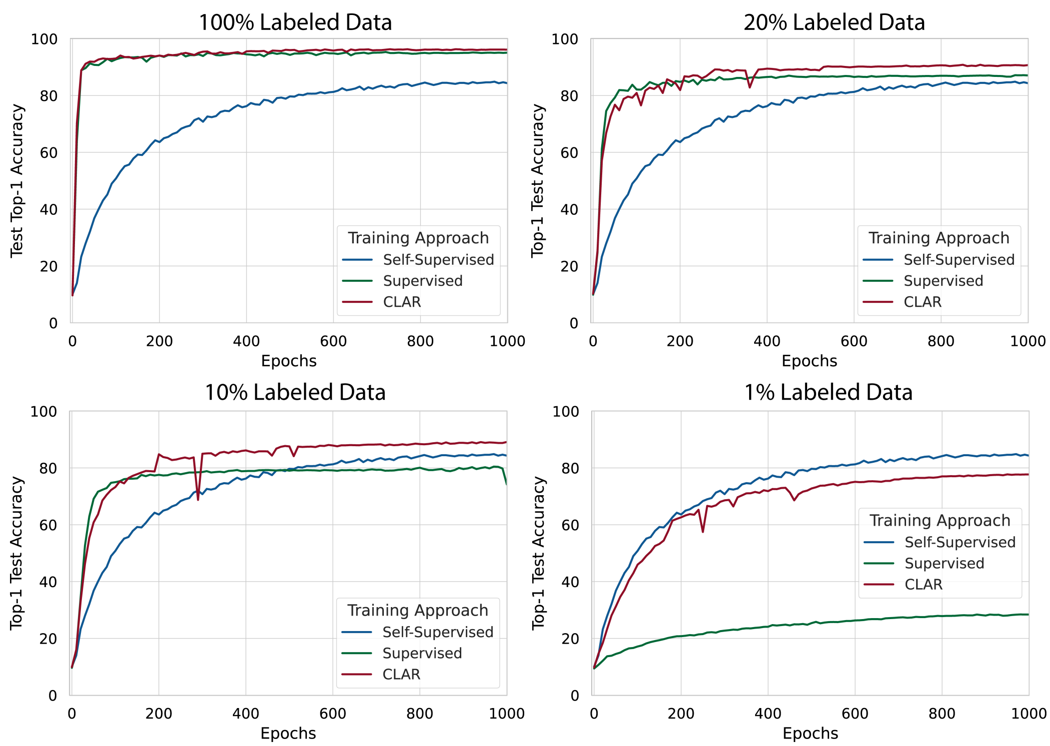


Figure 5: Top-1 test performance on Speech Commands-10 computed every 10-epochs by training an evaluation head attached to a frozen encoder over 1000 epochs. Each sub-figure represents the top-1 test performance on varying percentage of labeled data.

9 CONCLUSION

In this paper, we demonstrated the success of contrastive learning in earning efficient auditory representations. Our extensive and comprehensive experiments on various design choices revealed the effectiveness of our proposed framework (CLAR) in terms of recognition performance as well as reduced training time in comparison with supervised and self-supervised methods. Together, our results depicts a promising path towards automated audio understanding.

References

Allen, J. (1977). Short term spectral analysis, synthesis, and modification by discrete fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(3):235–238.

Bachman, P., Hjelm, R. D., and Buchwalter, W. (2019). Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15535–15545.

Baum, E. B. and Wilczek, F. (1988). Supervised learning of probability distributions by neural networks. In *Neural information processing systems*, volume 12, pages 52–61. New York, NY.

Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. (2019). Learning imbalanced datasets with label-distribution-aware margin loss.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Cheuk, K. W., Anderson, H., Agres, K., and Herremans, D. (2020). nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolutional neural networks. *IEEE Access*.

Choi, K., Fazekas, G., Sandler, M., and Cho, K. (2017). Convolutional recurrent neural networks for music classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396. IEEE.

Donahue, C., McAuley, J., and Puckette, M. (2019). Adversarial audio synthesis.

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks.

In *Advances in neural information processing systems*, pages 766–774.

Engel, J., Agrawal, K. K., Chen, S., Gulrajani, I., Donahue, C., and Roberts, A. (2019). GANSynth: Adversarial neural audio synthesis. In *International Conference on Learning Representations*.

Engel, J., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K., and Norouzi, M. (2017). Neural audio synthesis of musical notes with wavenet autoencoders.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. (2018). Accurate, large minibatch sgd: Training imagenet in 1 hour.

Guzhov, A., Raue, F., Hees, J., and Dengel, A. (2020). Esresnet: Environmental sound classification based on visual domain models. *ArXiv*, abs/2004.07301.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hénaff, O. J., Srinivas, A., De Fauw, J., Razavi, A., Doersch, C., Eslami, S., and Oord, A. v. d. (2019). Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*.

Hershey, S., Chaudhuri, S., Ellis, D. P., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., et al. (2017). Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE.

Hénaff, O. J., Srinivas, A., Fauw, J. D., Razavi, A., Doersch, C., Eslami, S. M. A., and van den Oord, A. (2019). Data-efficient image recognition with contrastive predictive coding.

Ibrahim, K. M., Royo-Letelier, J., Epure, E. V., Peeters, G., and Richard, G. (2020). Audio-based auto-tagging with contextual tags for music. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16–20.

Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., and Krishnan, D. (2020). Supervised contrastive learning.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R. S., Torralba, A., Urtasun, R., and Fidler, S. (2015). Skip-thought vectors.

Ko, T., Peddinti, V., Povey, D., and Khudanpur, S. (2015). Audio augmentation for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Kolesnikov, A., Zhai, X., and Beyer, L. (2019). Revisiting self-supervised visual representation learning.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Li, Z. and Hoiem, D. (2017). Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

Loshchilov, I. and Hutter, F. (2016). Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*.

McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

McFee, B., Humphrey, E. J., and Bello, J. P. (2015). A software framework for musical data augmentation. In *ISMIR*, volume 2015, pages 248–254.

McFee, B., Lostanlen, V., Metsai, A., McVicar, M., Balke, S., Thomé, C., Raffel, C., Zalkow, F., Malek, A., Dana, Lee, K., Nieto, O., Mason, J., Ellis, D., Battenberg, E., Seyfarth, S., Yamamoto, R., Choi, K., viktorandreevichmorozov, Moore, J., Bittner, R., Hidaka, S., Wei, Z., nullmightybofo, Hereñú, D., Stöter, F.-R., Friesch, P., Weiss, A., Vollrath, M., and Kim, T. (2020). librosa/librosa: 0.8.0.

Mittal, T., Bhattacharya, U., Chandra, R., Bera, A., and Manocha, D. (2020). Emotions don't lie: A deepfake detection method using audio-visual affective cues.

Oord, A. v. d., Li, Y., and Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019a). Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*.

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019b). Specaugment: A simple data augmentation method for automatic speech recognition. *Interspeech 2019*.

Piczak, K. J. (2015). ESC: Dataset for Environmental Sound Classification.

Qian, R., Meng, T., Gong, B., Yang, M.-H., Wang, H., Belongie, S., and Cui, Y. (2020). Spatiotemporal contrastive video representation learning.

Radhakrishnan, R., Divakaran, A., and Smaragdis, A. (2005). Audio analysis for surveillance applications. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2005.*, pages 158–161. IEEE.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Schlüter, J. and Grill, T. (2015). Exploring data augmentation for improved singing voice detection with neural networks. In *ISMIR*, pages 121–126.

Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., and Fergus, R. (2015). Training convolutional networks with noisy labels.

Tian, Y., Krishnan, D., and Isola, P. (2020a). Contrastive multiview coding.

Tian, Y., Sun, C., Poole, B., Krishnan, D., Schmid, C., and Isola, P. (2020b). What makes for good views for contrastive learning?

Tokozume, Y. and Harada, T. (2017). Learning environmental sounds with end-to-end convolutional neural network. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2721–2725. IEEE.

Tokozume, Y., Ushiku, Y., and Harada, T. (2017). Learning from between-class examples for deep sound recognition. *arXiv preprint arXiv:1711.10282*.

Warden, P. (2018). Speech commands: A dataset for limited-vocabulary speech recognition.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. (2018). Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742.

You, Y., Gitman, I., and Ginsburg, B. (2017). Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888*.

Zhuang, C., Zhai, A. L., and Yamins, D. (2019). Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6002–6012.