Contrastive Predictive Coding of Audio with an Adversary

Luyu Wang, Kazuya Kawakami, Aaron van den Oord

DeepMind

luyuwang@google.com, kawakamik@google.com, avdnoord@google.com

Abstract

With the vast amount of audio data available, powerful sound representations can be learned with self-supervised methods even in the absence of explicit annotations. In this work we investigate learning general audio representations directly from raw signals using the Contrastive Predictive Coding objective. We further extend it by leveraging ideas from adversarial machine learning to produce additive perturbations that effectively makes the learning harder, such that the predictive tasks will not be distracted by trivial details. We also look at the effects of different design choices for the objective, including the nonlinear similarity measure and the way the negatives are drawn. Combining these contributions our models are able to considerably outperform previous spectrogram-based unsupervised methods. On AudioSet we observe a relative improvement of 14% in mean average precision over the state of the art with half the size of the training data.

Index Terms: audio representations, audio tagging, unsupervised learning, self-supervised learning, adversarial attacks

1. Introduction

Learning general audio representations without explicit human supervision or labels is a challenging unsolved problem. As real-world audio signals have diverse noise conditions, a good model would need to disentangle signals from noise and learn meaningful representations for downstream tasks such as audio tagging [1, 2, 3, 4], sound event detection [5, 6], source separation [7, 8], etc. Self-supervised learning is an emerging area of research where models are developed to learn representations without labels or explicit supervisions. It leverages a proxy task to learn meaningful representations, whose quality benefits from massive amounts of unlabeled data. Previous works on image [9, 10, 11, 12, 13, 14, 15, 16], video [17, 18], and speech [19, 20, 21, 22] show that features learned with self-supervised learning improve data efficiency, transferability to other downstream tasks, and robustness to distribution shifts [21].

Contrastive Predictive Coding (CPC, [12]) is a self-supervised learning method that learns representations from a sequence by trying to predict future observations with a contrastive loss. The hypothesis is that the representations that predict the future well may encode semantically meaningful contents for downstream tasks. Although CPC has been widely applied to representation learning for different modalities, including image [14] and speech [19, 20, 21, 22], it is still a question whether this predictive objective can learn better representations for general non-speech audio, as diverse audio events may require very different features than speech data.

A common issue with self-supervised learning methods is that there often exist shortcuts to solving the proxy task in non-meaningful ways, resulting in representations that are irrelevant to downstream tasks. For example, the model may pick up on certain traits of the noise in the audio instead of the object or event making the sound. Recently, pre-defined data augmentations have been used to guide the self-supervised learner to focus on signals relevant to downstream tasks instead of irrelevant factors (e.g. rotation or blur for images), but usually one needs to handcraft and combine various types to get good results [13, 14, 15, 16]. Although they are effective at making the model invariant to certain low-level features which could help defend against shortcuts, such augmentations require prior knowledge about the data and the downstream task to be useful.

In this paper, we propose to learn audio representations with the CPC objective. We use the resulting representations for the audio tagging problem on AudoSet [23] to evaluate whether they are correlated with the underlying audio events. We show that the CPC approach, which operates directly on raw audio, can provide substantial improvements over previous spectrogram-based audio models [2, 3]. We also investigate two extensions. First, we improve contrastive learning mechanism of CPC using nonlinear similarity functions. This is previously proposed in SimCLR [16], yet whether it is applicable to the CPC objective remains unknown. Moreover, we apply adversarial perturbations [24, 25, 26] to audio signals in order to avoid trivial solutions during the self-supervised learning. We found that being invariant to the adversarial noise can further improve the quality of representations. Unlike manual augmentations used in previous works, our adversarial perturbations can be used when there is no prior knowledge about data. To our knowledge, this is the first time adversarial perturbations are shown to be able to improve self-supervised learning.

2. Method

In this section, we first briefly review the CPC prediction task, and then introduce proposed extensions. For more details on CPC we refer to the original paper [12].

2.1. Contrasting predictive coding

Self-supervised training of neural networks depends on the model solving a proxy task, in the expectation that the features learned are discriminative for any downstream tasks that require labels. On audio, the proxy task defined by CPC is to predict the future steps of the clip in latent space given the summary of the past contexts encoded with an autoregressive model.

More specifically, given a raw audio signal $\mathbf{x} = (x_1, x_2, \ldots, x_L, x_i \in \mathbb{R})$, an encoder network $f(\cdot)$ encodes it into vector representations called the latents $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_M, \mathbf{z}_i \in \mathbb{R}^{d_z})$, where d_z is the dimensionality of the latent space, and M corresponds to the length of the sequence of latent vectors. Then an autoregressive network $g(\cdot)$ encodes the latent vectors and produces contextualized vectors $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_M, \mathbf{c}_i \in \mathbb{R}^{d_c})$, where $\mathbf{c}_i = g(\mathbf{z}_{\leq i})$. The task is then to predict the k-th step into the future denoted as $h_k(\mathbf{c}_i)$. The loss is formulated in the way that $h_k(\mathbf{c}_i)$ should resemble the k-th step latent vector $\mathbf{z}_{i+k}$, but be dissimilar to the others which we call negatives $\tilde{\mathbf{z}}$.

Instead of using a softmax classification objective that includes all possible latent vectors in the batch as negatives [12], we use a sampling objective which samples negatives from a proposal distribution p_n:

$$L_k = -\sum_i \Big(\log \sigma \left(\text{sim} \left(\mathbf{z}_{i+k}, h_k(\mathbf{c}_i) \right) / \tau \right) + \lambda \mathop{\mathbb{E}}_{\tilde{\mathbf{z}} \sim p_n} \left[\log \sigma \left(-\text{sim} \left(\tilde{\mathbf{z}}, h_k(\mathbf{c}_i) \right) / \tau \right) \right] \Big) \tag{1}$$

where σ denotes the sigmoid function, τ stands for the temperature parameter, and $h_k(\cdot)$ is an affine transformation. The expectation is approximated by the empirical average and λ is set as the number of negative samples. The similarity measure originally is the dot product which takes the form of $\text{sim}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^T \mathbf{v}$. We optimize the total prediction loss $L = \sum_{k=S}^{K} L_k$, where S and K are the start and end step to predict, respectively. Note that it is important to skip a few steps to avoid overlapping between the receptive field of $\mathbf{z}_{i+k}$ and $\mathbf{c}_i$ on the input signal, otherwise the information in the former can be leaked to the latter, which results in trivial solutions to the objective.

We investigate with three different ways to define the proposal distribution p_n: uniformly sample negatives from different locations in the same audio signal (**local sampling**), from other signals in a mini-batch (excluding the current signal) (**exclusive sampling**), or from all samples in the mini-batch (including the same audio signal) (**cross sampling**).

2.2. Nonlinear learnable similarity metric

In the recent SimCLR model [16], it is shown that using a nonlinear transformation together with a cosine similarity function (instead of using simple dot product similarity) in the contrastive loss is beneficial for the downstream performance in the linear classifier task for images. Following the same strategy the similarity function in Equation 1 becomes

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{p(\mathbf{u})^T p(\mathbf{v})}{\|p(\mathbf{u})\| \, \|p(\mathbf{v})\|} \tag{2}$$

where $p(\cdot)$ is a multi-layer perceptron (MLP) with one hidden layer. Note that tuning the temperature parameter τ is crucial to make this nonlinear cosine similarity measure work.

2.3. Augmentation with adversarial perturbations

In practice, predicting the k-th future latent step from $\mathbf{c}_i$ can be easier using trivial and non-semantic information rather than meaningful contents of underlying audio events. In order to avoid such shortcut solutions, we may augment the input audio with additive Gaussian noise, speed perturbation, or pitch shift similarly to data augmentation strategies used in previous self-supervised learning works on image [13, 14, 15]. However, it takes lots of prior knowledge and experimentation to handcraft the appropriate composition [16]. Thus, we proposed to *learn* the augmentation through adversarial perturbations to make the proxy task harder. This resembles adversarial training in supervised learning [25, 27], which can avoid learning highly predictive but incomprehensible features [28]. We expect being invariant to such small perturbations can prevent the self-supervised model from focusing on trivial signals in the data.

We consider the simplest form of adversarial perturbations as the first attempt in this direction. The so-called fast gradient sign method (FGSM) [25] is defined as:

$$\mathbf{x}_{adv} = \mathbf{x} + \epsilon \text{sign}(\nabla_{\mathbf{x}} L) \tag{3}$$

Algorithm 1: Training CPC with adversarial perturbations.

input: batch size N, constant τ, constant ϵ, encoder network f, context network g
for minibatch $\{\mathbf{x}_k\}_{k=1}^{N}$ **do**
  $\mathbf{z}_k = f(\mathbf{x}_k)$
  for $i \in \{1, \ldots, M\}$ **do**
    $\mathbf{c}_i^k = g(\mathbf{z}_{\leq i}^k)$
  end
  $\tilde{\mathbf{z}} \sim \mathbf{z}_k$ # assume local sampling
  $\mathbf{x}_k^{adv} = \mathbf{x}_k + \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}_k} L(\mathbf{c}_k, \mathbf{z}_k, \tilde{\mathbf{z}}, \tau))$
  $\mathbf{z}_k^{adv} = f(\mathbf{x}_k^{adv})$
  $\tilde{\mathbf{z}}^{adv} \sim \mathbf{z}_k^{adv}$
  update f and g to minimize $L(\mathbf{c}_k, \mathbf{z}_k^{adv}, \tilde{\mathbf{z}}^{adv}, \tau)$
end
return trained networks f and g

where ϵ is a hyperparameter controlling the magnitude of the perturbation step in the input space. Given $\mathbf{x}_{adv}$, now we use the same context $\mathbf{c}_i$ to predict instead the augmented latents $\mathbf{z}_{adv} = (\mathbf{z}_1^{adv}, \mathbf{z}_2^{adv}, \ldots, \mathbf{z}_M^{adv}, \mathbf{z}_i^{adv} \in \mathbb{R}^{d_z})$, from which the negatives are also sampled. The steps are detailed in Algorithm 1. Notice the sign operation stops the gradients from getting into the perturbations part. An advantage of using FGSM is that it only requires one more full backward pass during training, compared to other methods of producing adversarial examples that may require multiple gradient steps.

3. Experimental setup

3.1. Model architecture

The encoder network $f(\cdot)$ consists of 10 one-dimensional convolutional layers, with kernel sizes (10, 4, 4, 4, 4, 4, 4, 4, 1, 1), strides (5, 2, 2, 2, 2, 2, 2, 2, 1, 1), and channel size 512. We extract audio patches of 40 ms for every 20 ms for the encoder. The input audio sampling frequency is 16 kHz and the encoder reduces it to 50 Hz. We use the Swish nonlinearity [29] and observe it better stabilizes the training than ReLU. The context network $g(\cdot)$ has 12 stride-1 causual convolutional layers with ReLU activations and kernel sizes increasing from 2 to 13. Dense connections [30] are used in the context net as it improves convergence. In total the context covers a receptive field of 1.62 second. We use two context networks to compute CPC objectives both from past to future and the other way around. Modeling bidirectional relationships in CPC has previously been shown to be very effective [21]. Layer normalization layers are used after each convolutional layer in both the encoder and context network.

To compute the objective, we sample 2 negatives for each of the $K = 4$ tasks. We skip the first 3 predictions, to avoid overlapping between the latent and the context representations on the inputs, otherwise the latter can "see" part of the near future such that it can "cheat". For the nonlinear projection head $p(\cdot)$ we use a MLP with a single hidden layer of size 512. A temperature of 0.05 is used for our CPC objective.

The models are implemented in TensorFlow v1, and trained using a total batch size of 256 on the third generation tensor processing units (TPUs) with 32 cores for up to 400k steps. Adam optimizer [31] is used for optimization, starting with a learning rate of 10^{-4}, and then gradually decaying it following the cosine curve down to 10^{-6} [32].

Table 1: *Effects of the FGSM magnitude (ϵ) on the CPC pre-training accuracy and the mean average precision of the downstream shallow model classification on AudioSet. The training CPC accuracy is measured on the augmented training data, and the holdout accuracy is on the clean development set.*

FGSM magnitude (dB)	CPC accuracy		AudioSet
	train	dev	dev mAP
-75	0.966	0.895	0.271
-65	0.964	0.890	0.272
-55	0.963	0.886	**0.275**
-45	0.964	0.876	0.270
-35	0.967	0.865	0.267
No augmentation	0.970	0.905	0.268
Additive Gaussian	0.968	0.901	0.270

3.2. Evaluation

We use the audio segments from AudioSet [23] for both training and evaluation. It consists of over 2 million 10-second clips from YouTube, each manually annotated with one or more labels from an ontology of 527 categories. We use the publicly available version of the unbalanced training set (41% smaller than that used in the baseline systems [2, 3]). We split the original training set into training and development subset by 95% and 5%, respectively. We found in our experiments it is better to take the full 10-seconds audio as inputs rather than training on shorter segments.

We evaluate the representations in the downstream task of training shallow fully connected audio classifiers following the same setup as in [2, 3], where a 1-hidden-layer MLP with 512 units is used. We average over the time dimension of the final embeddings of the context network and then feed them into the shallow models. The classifiers are trained on the frozen CPC features, and a sigmoid classification loss is used to account for the multi-instance nature of the audio tagging problem. In our experiments we found that the context features $\mathbf{c}$ always perform better than latents $\mathbf{z}$ in the downstream task, possibly because the former has a larger receptive field from the extra neural network layers.

4. Results

In this section we detail some of key factors that affect the model performance based on the development set. Then we show how the proposed method compares to the state of the art on the test set. Since the CPC objective in Equation 1 solves a classification problem (distinguishing positives from negatives), we use the accuracy to assess its difficulty. We measure both the training and heldout CPC accuracy, and the latter is calculated without any augmentation on the input data. For the downstream tasks we report the mean average precision (mAP) score.

4.1. Effects of the adversarial perturbations

In Table 1, we show how adversarial augmentation affects the CPC training as well as the downstream performance. On one hand, when the perturbation magnitude is too small, the CPC accuracy is close to that without any augmentation. On the other hand, when the magnitude is too big, the perturbation itself becomes a trivial feature which makes the inputs less natural: it is easier for the CPC prediction tasks, but does not capture the underlying semantic information well. This is evident by the high training but low holdout CPC accuracy when $\epsilon = -35$ dB, which results in a poor downstream performance. When the perturbation strength is around -55 dB, the training accuracy is the lowest, meaning that the attack is strong and the tasks become harder, and it requires the CPC network to extract more essential sound contents. As a result, it achieves the highest mAP on the downstream task. It significantly outperforms counterparts without any augmentation or with additive Gaussian noise. In our experiments we find Gaussian noise of different standard deviations result in similar performance (unless it makes the noise overwhelmingly strong which hurts the performance). Adding this kind of noise distracts the model from some high frequency trivial features, as we see a slightly better performance on the downstream task. However, it does not produce predictive tasks as hard as adversarial augmentations.

Table 2: *Effects of sampling methods on the CPC pre-training accuracy, the mean average precision of the downstream shallow model classification on AudioSet, and the accuracy on the DCASE2013 audio classification task.*

Sampling	CPC accuracy		AudioSet	DCASE2013
	train	dev	dev mAP	test accuracy
Local	0.906	0.899	0.267	0.94
Cross	0.970	0.905	**0.268**	0.95
Exclusive	0.994	0.994	0.240	**0.98**

4.2. Negative sampling distributions

Previously, it was shown for speech recognition, sampling negatives within the same sample is always better than including those from other ones [19, 21]. One explanation is that this makes CPC representations focus on linguistic features rather than nonspeech traits such as the speaker identity or background noise, as we can not use the latter to transcribe the speech. However, this intuition does not seem to hold for general audio downstream tasks including audio tagging, in which being able to discriminate such features is important.

Therefore, we conduct experiments on three sampling strategies introduced in Section 2.1 and results are shown in Table 2. We use the same number of negatives to have a fair comparison on the accuracy. Sampling from the same clip only (local sampling) results in the worst training CPC accuracy. Intuitively, in this case the predictive tasks are hard because the negatives share many similar traits with the positives. Including negatives from other samples (cross sampling) makes the CPC task easier, evident by a higher training accuracy. However, it does not make a big difference on the downstream task. It might be because with this sampling strategy the hard negatives are dominated by local ones which produce the most training signals. On the contrary, only sampling from other samples (exclusive sampling) makes it so easy that the CPC network is not learning temporal proximity information well, but it does learn how to tell the differences between the overall traits in one sample from those in others: interestingly, it does not perform well on the audio tagging task on AudioSet, but it achieves the new state of the art on DCASE2013 test set with a linear classifier [33, 34].

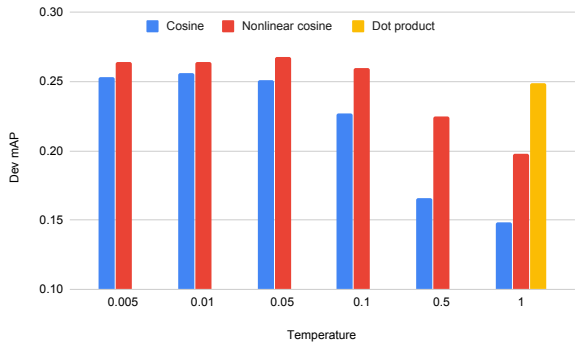

Figure 1: *Impact of the temperature on different similarity measures.*

4.3. Different similarity measures

Similar to the findings in [16], we notice there is a noticeable gain from using the nonlinear cosine similarity measure. Figure 1 shows that it is important to use an appropriate temperature parameter. When it is 1, the dot product used in the original CPC formulation has the best result; when it is 0.05 this learnable nonlinear measure stands out.

4.4. Comparison to the state of the art

Based on the best performing setting on the development set, we train our models up to one million steps and evaluate them on the test set. Table 3 shows that our CPC-based model achieves a substantial improvement over previous baselines. Notice that the size of our training data is 59% of what is used for the triplet [2] and coincidence, categorization, and consolidation (C^3) [3] baseline models in Table 3. Trained on raw audio without any augmentation, CPC outperforms the previous best models which operate on heavily augmented spectrograms. Adding adversarial noise further improves the performance.

Recent works on cross-modal self-supervised training have shown that grounding in video can significantly improve the performance of the audio representations. Yet the proposed method is able to outperform the Look, Listent, and Learn (L^3) [17] benchmark without requiring video. Our approach can find use cases where there is only audio data available, or when the storage and computation requirements for the multimodal training are unaffordable.

5. Related work

There have been several self-supervised learning methods developed recently for general audio representations. In [2], the triplet loss is used to make pairs of audio segments agree with each other under temporal proximity, small additive Gaussian noise, time/frequency translation, and example mixing on the spectrograms. An approach inspired by learning language models is explored for mobile audio applications in [35]. Researchers have increasingly been interested in multimodal training objectives to learn better audio representations. Look, Listen, and Learn [17, 34] learns by predicting audio-viual frame correspondence. Coincidence, Categorization, and Consolidation [3] combines a unimodal or cross model coincidence objective, a clustering objective, and cluter-based active learning process to achieve the state-of-the-art performance. Note that

Table 3: *Performance of shallow model classification with fixed representations on AudioSet test subset.*

Model	Train input	test mAP
Using video		
L^3 net [17]	spectrogram & video	0.249
C^3 net [3]	spectrogram & video	**0.285**
Audio only		
C^3 net [3]	spectrogram	0.206
Triplet [2]	spectrogram	0.226
Triplet [2]	spectrogram (+ augmentation)	0.244
CPC (ours)	raw audio	0.267
CPC (ours)	raw audio (+ adversarial noise)	**0.277**

all of them are based on spectrograms whereas the proposed method operates on raw audio directly.

CPC for speech recognition has recently been actively studied. In [19] the authors show that pretrained CPC frontends can serve as a drop-in replacement for traditional spectrograms and significantly decrease the word error rate of supervised recognition systems. A vector quantized extension is proven to further improve the performance [20]. Later in [21, 22] it is discoverd that CPC benefits from having more unlabeled pretraining data from a variety of different sources, resulting in more robust representations that transfer across different languages. Although speech is just one form of audio, recognizing it requires the representations to focus on the linguistic features while ignoring other acoustic contents.

Orthogonal to self-supervised learning, it is known that supervised models are subjected to adversarial vulnerabilities [24, 25, 26, 36], including those for speech [37, 38] and non-speech audio [39]. So far, the most successful way of mitigating this issue is through adversarial training [27, 28].

6. Conclusions

In this work, we apply the CPC framework to learning general audio representations. We investigate the use of adversarial perturbations and the nonlinear learnable similarity measure to improve the CPC learning. We also look at the effects of different design choices. Our results show both extensions to the original CPC formulation increase the downstream performance on the audio tagging problem on AudioSet. Our CPC-based model is able to bridge the gap between unsupervised representations trained with audio only and with both audio and video.

7. Acknowledgements

We would like to thank Karen Simonyan and the anoynomous reviewers for help feedback.

8. References

[1] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "Cnn architectures for large-scale audio classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 131–135.

[2] A. Jansen, M. Plakal, R. Pandya, D. P. Ellis, S. Hershey, J. Liu, R. C. Moore, and R. A. Saurous, "Unsupervised learning of semantic audio representations," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 126–130.

[3] A. Jansen, D. P. Ellis, S. Hershey, R. C. Moore, M. Plakal, A. C. Popat, and R. A. Saurous, "Coincidence, categorization, and consolidation: Learning to recognize sounds with minimal supervision," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 121–125.

[4] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *arXiv preprint arXiv:1912.10211*, 2019.

[5] Q. Kong, Y. Xu, I. Sobieraj, W. Wang, and M. D. Plumbley, "Sound event detection and time–frequency segmentation from weakly labelled data," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 4, pp. 777–787, 2019.

[6] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 31–35.

[7] E. Tzinis, S. Wisdom, J. R. Hershey, A. Jansen, and D. P. Ellis, "Improving universal sound separation using sound classification," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 96–100.

[8] Q. Kong, Y. Wang, X. Song, Y. Cao, W. Wang, and M. D. Plumbley, "Source separation with weakly labelled data: An approach to computational auditory scene analysis," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 101–105.

[9] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with convolutional neural networks," in *Advances in neural information processing systems (NeurIPS)*, 2014, pp. 766–774.

[10] C. Doersch, A. Gupta, and A. A. Efros, "Unsupervised visual representation learning by context prediction," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1422–1430.

[11] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 69–84.

[12] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[13] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 15 509–15 519.

[14] O. J. Hénaff, A. Razavi, C. Doersch, S. Eslami, and A. van den Oord, "Data-efficient image recognition with contrastive predictive coding," *arXiv preprint arXiv:1905.09272*, 2019.

[15] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," *arXiv preprint arXiv:1911.05722*, 2019.

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

[17] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 609–617.

[18] A. Owens and A. A. Efros, "Audio-visual scene analysis with self-supervised multisensory features," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 631–648.

[19] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *Proc. Interspeech 2019*, pp. 3465–3469, 2019.

[20] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," *International Conference on Learning Representations (ICLR)*, 2019.

[21] K. Kawakami, L. Wang, C. Dyer, P. Blunsom, and A. van den Oord, "Learning robust and multilingual speech representations," *arXiv preprint arXiv:2001.11128*, 2020.

[22] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," *arXiv preprint arXiv:2002.02848*, 2020.

[23] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "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.

[24] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *International Conference on Learning Representations (ICLR)*, 2014.

[25] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[26] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.

[27] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *International Conference on Learning Representations (ICLR)*, 2018.

[28] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 125–136.

[29] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," *arXiv preprint arXiv:1710.05941*, 2017.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[32] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *International Conference on Learning Representations (ICLR)*, 2017.

[33] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and classification of acoustic scenes and events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[34] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3852–3856.

[35] M. Tagliasacchi, B. Gfeller, F. d. C. Quitry, and D. Roblek, "Self-supervised audio representation learning for mobile devices," *arXiv preprint arXiv:1905.11796*, 2019.

[36] G. W. Ding, K. Y. C. Lui, X. Jin, L. Wang, and R. Huang, "On the sensitivity of adversarial robustness to input data distributions," *International Conference on Learning Representations (ICLR)*, 2019.

[37] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 2018, pp. 1–7.

[38] Y. Qin, N. Carlini, I. Goodfellow, G. Cottrell, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *International Conference on Machine Learning (ICML)*, 2019, pp. 5231–5240.

[39] V. Subramanian, A. Pankajakshan, E. Benetos, N. Xu, S. McDonald, and M. Sandler, "A study on the transferability of adversarial attacks in sound event classification," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 301–305.