TRICYCLE: AUDIO REPRESENTATION LEARNING FROM SENSOR NETWORK DATA USING SELF-SUPERVISION

Mark Cartwright[1*], *Jason Cramer*[1], *Justin Salamon*[2] , *Juan Pablo Bello*[1]

[1] Music and Audio Research Laboratory, New York University, NYC, NY, USA
[2] Adobe Research, San Francisco, CA, USA

[*]Please address correspondence to Mark Cartwright (mark.cartwright@nyu.edu)

ABSTRACT

Self-supervised representation learning with deep neural networks is a powerful tool for machine learning tasks with limited labeled data but extensive unlabeled data. To learn representations, self-supervised models are typically trained on a pretext task to predict structure in the data (e.g. audio-visual correspondence, short-term temporal sequence, word sequence) that is indicative of higher-level concepts relevant to a target, downstream task. Sensor networks are promising yet unexplored sources of data for self-supervised learning—they collect large amounts of unlabeled yet timestamped data over extended periods of time and typically exhibit long-term temporal structure (e.g., over hours, months, years) not observable at the short time scales previously explored in self-supervised learning (e.g., seconds). This structure can be present even in single-modal data and therefore could be exploited for self-supervision in many types of sensor networks. In this work, we present a model for learning audio representations by predicting the long-term, cyclic temporal structure in audio data collected from an urban acoustic sensor network. We then demonstrate the utility of the learned audio representation in an urban sound event detection task with limited labeled data.

Index Terms— self-supervised learning, representation learning, audio embedding, sensor network

1. INTRODUCTION

Honking at rush hour, the clang of garbage trucks on Monday mornings, children playing after school, church bells at noon, Shabbat sirens on Fridays, summer concerts in the park, the dawn chorus of birds: these are all examples of the sounds that define city soundscapes and represent the spatio-temporal structure of city life. In this paper, we leverage this structure using self-supervision to learn audio representations from an urban acoustic sensor network, SONYC [1]. We then use these representations as the input to a downstream urban sound tagging task.

Self-supervised learning [2, 3] is a machine learning paradigm in which representations are learned by training models on proxy, "pretext", tasks using an abundance of unlabeled data, with the goal of increasing data efficiency in downstream tasks (e.g., using less labeled data to achieve comparable performance on a supervised task). Successful pretext tasks are those that require some understanding of the concepts of interest in the downstream task. Therefore, a common pretext task is to predict structural aspects of data, e.g., sequential structure [4], spatial structure [3], short-term temporal structure (scale of seconds) [5, 6, 7, 8], or multi-modal structure [9, 10, 11, 12, 13, 14, 15].

In this work, we propose TriCycle, a neural network model to learn audio representations by predicting the long-term temporal structure of data, a novel pretext task that has not previously been explored. TriCycle uses the timestamp of an audio recording as a relative measure of temporal structure which is folded into three cycles: day, week, and year. It then predicts the phase of these cycles from the audio and recording location, and we use feature maps of the neural network as an input representation in downstream tasks. For a downstream task in the same domain as the pretext task, TriCycle performs comparably to L^3, a state-of-the-art audio representation trained with multi-modal supervision [13, 14].

While multi-modal self-supervision (e.g., audio-visual correspondence (AVC) and synchronization) has been successfully applied to several audio tasks [12, 11, 10, 9, 13, 15, 14], single-modal supervision has been relatively unexplored for audio representation learning [16, 17]. Yet, there are plenty of audio domains that have long-term temporal structure but whose higher-level concepts do not have clear visual correlates (e.g. machine condition monitoring) or consist of far-field events that are difficult to catch on camera (e.g., bioacoustic monitoring [18]). The supervision in TriCycle may be more suited than AVC at capturing concepts of interest for downstream tasks in those domains. Since it eliminates the need for visual data, TriCycle can also be used when multimodal data is not captured due to privacy concerns (e.g., in urban sound monitoring).

This supervision (temporal cycle prediction) seems particularly well-suited to sensor network data, which is dense in time and longitudinal. While we apply this approach to representation learning on data collected from an urban acoustic sensor network, this approach may not be limited to audio. This self-supervision could possibly be used to learn representations from other large sets of unlabeled, yet timestamped data, e.g., data from non-audio sensor networks, which are becoming ubiquitous with the rise of IoT and smart sensing in cities [19].

2. SONYC ACOUSTIC SENSOR NETWORK DATA

The Sounds of New York City (SONYC) is a project to monitor, analyze, and mitigate urban noise pollution [1] through smart urban sensing. As of May 2019, the project has deployed 57 acoustic sensors in New York City on the outside of buildings and poles at a height of 20 feet. The sensors have collectively recorded approximately 40 years of audio data since 2016 in the form of 130M 10 s recordings, which are unlabeled except for the time they were recorded and the sensor they were recorded from. Due to resource

[*]This work was partially supported by NSF awards 1544753 and 1633259.

constraints when training the models, we limit this data to that from 20 sensors recorded in 2017, resulting in 23.5M 10 s recordings. The sensors also collect A-weighted sound-pressure level (SPL in dBA) readings every 0.125 s.

We use the audio recordings with the timestamp supervision to train the TriCycle model, and the SPL readings to aid in sampling eventful recordings. In addition, as part of the 2019 Detection and Classification of Acoustic Scenes and Events (DCASE) Challenge Task 5 [20], we asked citizen science volunteers on the Zooniverse platform [21] to annotate the presence of 23 urban sound event classes from eight class groups (*engine*, *machinery impact*, *non-machinery impact*, *powered saw*, *alert signal*, *music*, *human voice*, *dog*) for 2,794 of the recordings. Three volunteers annotated each recording, and we aggregated their responses using minority vote, which we have found in previous studies to produce aggregate labels with high recall while still maintaining precision [22]. We use these aggregate labels for both training and testing in our downstream urban sound tagging task. In this work, we only use the 8 group level labels. We refer to this dataset as SONYC Urban Sound Tagging (SONYC-UST).

3. MODEL AND TRAINING

The TriCycle model is trained by predicting the time at which the input audio example was recorded. We do this by converting time into phase on three different temporal cycles: *day*, *week*, *year*. The model assumes that audio examples that are recorded at similar phases in these cycles are more likely to contain similar audio events. While not true for all audio recordings, intuition regarding the nature of sound in cities and preliminary analysis of our sensor data indicate that this is a reasonable assumption for outdoor urban acoustic sensor recordings.

We encode these temporal cycles into response variables by computing the sine and cosine of the phase within a time cycle (note the phase in radians, e.g. $\phi_d = 2\pi(\text{hour}/24)$ for the day cycle). Computing mean square error on these response variables, leads to the following loss function and optimization objective:

$$\ell(\hat{Y}, \Phi) = \sum_{c \in \{d,w,y\}} \frac{1}{2} \left((\sin(\phi_c) - \hat{y}_{c,1})^2 + (\cos(\phi_c) - \hat{y}_{c,2})^2 \right) \tag{1}$$

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^{n} \ell(f(X_i|\theta), \Phi_i) \tag{2}$$

where X_i is the input time-frequency representation for an audio clip, $\Phi = [\phi_d, \phi_w, \phi_y]$ is the vector of the phases of the day, week, and year temporal cycles, $[\hat{y}_d, \hat{y}_w$ and $\hat{y}_y]$ are the corresponding model predictions of phase, and θ is the model parameters.

The model architecture can be decomposed into two parts: an *audio subnetwork* and a *temporal prediction network* (see Figure 1). We borrow the audio subnetwork architecture from the *Look, Listen and Learn* (L^3-Net) model [14]. This architecture is composed of 4 convolutional blocks each of which contains 2 layers of 3×3 convolutional filters (64, 128, 256 and 512 filters per layer in each block respectively), followed by a 2×2 max-pooling layer with stride of 2. Each individual convolutional layer is also followed by batch normalization [23] and ReLU activations [24]. The subnetwork is max-pooled across the frequency and spatial dimension to produce a 512 dimensional (512-D) feature vector. This is the representation we extract for downstream tasks.

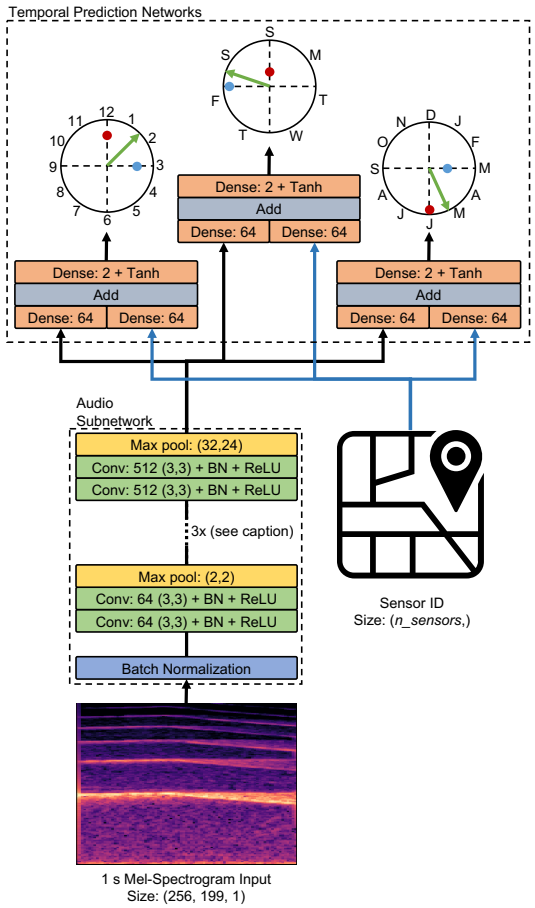


Figure 1: The TriCycle model, trained to predict the timestamp of the input, encoded as $sin(\phi_c)$ and $cos(\phi_c)$ where ϕ_c the phase in a day, week, and year. These values are represented by blue and red dots within the circles. The audio subnetwork consists of 4 blocks of convolution layers (only 2 shown). See Section 3 for details.

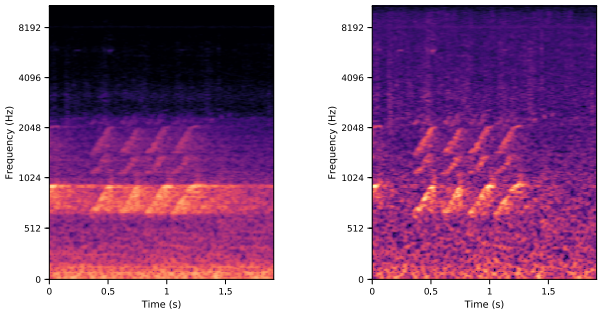
The temporal prediction network is composed of 3 simple subnetworks, each predicting a different temporal cycle. All of them receive the output of the audio subnetwork and the sensor ID, encoded as a 1-hot vector, as input. For each temporal subnetwork, the input from the audio subnetwork is processed by a 64-unit dense layer without a bias term, and the sensor ID is processed by a 64-unit dense layer. The outputs of these two layers are then added together and processed through a 2-unit dense output layer with $\tanh$ activations to predict the $\sin$ and $\cos$ of the phases. The sensor ID input was included to model differences in sound event distributions by spatial location.

The input to the model is 1 s of audio sampled at 48 kHz. While the original L^3-Net audio subnetwork transformed the time-domain audio input using a linear-frequency log-magnitude spectrogram (0.01 s windows with 50% overlap, 257 frequency bands), it has been found that an L^3-Net embedding trained with 256-band mel-frequency spectrograms input achieved higher performance on downstream environmental sound classification tasks [13]. We adopt the latter input representation in this work.

3.1. Sampling

We investigated two sampling methods: *random* and *high-activity*. Using random sampling, we randomly select a 10 s audio recording from the data set, and then randomly select a 1 s clip from the recording. Using high-activity sampling, we aim to focus the training on informative examples containing audio events. To detect events efficiently at scale, we limit this computation to the SPL data, and we compute the L_2 norm of the discrete difference of SPL in time, e.g. $\sqrt{\sum_{n=0}^{79}(d_{m,n} - d_{m,n-1})^2}$ for SPL sequence d of length 80 (i.e. 10 s with 0.125 s hop size) from sensor m. This assumes that audio events of interest manifest as large changes in SPL. To evenly sample in time, we randomly select an audio recording from the top 15th percentile of these values within every hour in the year for each sensor, effectively limiting our pool to 3.5M recordings. Within each recording, we randomly select a 1 s clip weighted by SPL.

3.2. Per-Channel Energy Normalization



(a) Log scaling (b) PCEN

Figure 2: A mel-frequency spectrogram of a siren pre-processed by log scaling and PCEN.

To further focus the model on events of interest and increase robustness to variation in acoustic background and sensors, we also investigated using Per-Channel Energy Normalization (PCEN) [25] pre-processing as an alternative to log-scaling the mel-frequency spectrogram inputs. PCEN aims to improve robustness to channel distortion by combining dynamic range compression and adaptive gain control with temporal integration:

$$\mathbf{PCEN}(t,f) = \left(\frac{\mathbf{E}(t,f)}{(\varepsilon + (\mathbf{E} \overset{t}{*} \boldsymbol{\phi}_T)(t,f))^{\alpha}} + \delta\right)^{r} - \delta^{r} \quad (3)$$

where ϕ_T is a low-pass filter at a time scale T and $\mathbf{E}(t,f)$ is the mel-frequency spectrogram. Starting with the guidelines in [26], we manually tuned the free parameters ($\alpha = 0.7$, $\varepsilon = 10^{-10}$, $r = 0.125$, $T = 0.25$, and $\delta = 0$) to decorrelate the frequency bands and "whiten" the background while preserving the sound events of interest (see Figure 2 for an example).

4. EXPERIMENTS

We evaluate TriCycle on two downstream tasks: 1) multi-label urban sound tagging and 2) sensor classification. The urban sound tagging task is our primary task of interest, whereas the sensor classification task may inform us if the embedding model is overfitting. For example, a large improvement in sensor classification performance combined with a decrease in urban sound tagging performance indicates that the model is overfitting to the acoustic characteristics of the sensors and background.

We compared variations of the TriCycle embedding to the L^3-Net embedding as input representations on our two downstream tasks. Since TriCycle and L^3-Net have the same audio subnetwork architecture, we also compared a randomly initialized audio subnetwork model as a feature extractor to establish the gain in performance from the two training paradigms. The baseline TriCycle model has randomly initialized weights, a log-magnitude mel-frequency spectrogram input, and high-activity sampling during training. We tested 4 additional variations. The first was trained with random sampling instead of high-activity sampling. The second preprocessed the mel-frequency spectrograms with PCEN instead of log-scaling. The third and fourth initialize the weights to those of L^3-Net to investigate if TriCycle can refine a general embedding using single-modal data matched to the downstream task. This third variation was refined with TriCycle using the same initial learning rate as the baseline (10^{-5}), and the fourth was refined using a lower learning rate (10^{-6}). Each of these variants will henceforth be referred to by the names specified in Table 1.

The Tricycle models were trained for 1500 epochs of 16k training examples using the Adam optimizer. Before training, a validation set of 16k was removed from the training set, and any recordings within 10 s of the validation set recordings were also removed from the training set. The best model on the validation set was used in the downstream tasks. To gain further insight on how the models perform on the pretext task, we look at the median angle between the predicted phases and the ground truth phases for each cycle, which we refer to as *median angular displacement* (MAD).

The downstream models for the urban sound tagging task were trained and evaluated using the SONYC-UST dataset, which contains 10 s audio recordings sampled at 48kHz. To test the generalizability of the embeddings to new sensors, we split SONYC-UST by sensor into a training set of the same 20 sensors used when training the the TriCyle embeddings (1952 recordings) and a test set of the remaining 23 sensors (842 recordings). Embeddings were extracted at 100 ms intervals.

We trained a shallow model containing a dense layer with eight sigmoid outputs, each corresponding to the presence of one of the eight labels. The embedding for each frame is input into the model and their outputs are aggregated using AutoPool [27] to produce a recording-level output. The models were trained for 1500 epochs to minimize binary cross-entropy for each label in the recording-level output, using the Adam optimizer with an initial learning rate of 10^{-3} and a weight decay factor of 10^{-5}. We evaluated each multi-label classification model on the test set using micro-averaged F1-score, precision, and recall using a threshold of 0.5, as well as the micro-averaged area under the precision-recall curve (AUPRC).

The downstream models for the sensor classification task were also trained and evaluated using the SONYC-UST dataset. For this task, we split SONYC-UST into 10 stratified folds. Again, embeddings were extracted at 100 ms intervals. We trained a shallow model containing one dense layer with 43 outputs with a softmax non-linearity applied, corresponding to a multi-class prediction for one of the sensors. The embedding for each frame is input into the model, producing a distribution of sensors for each frame. The models were trained for 1500 epochs to minimize categorical cross-entropy for each the output prediction of each individual frame, using the Adam optimizer with an initial learning rate of 10^{-3} and a

	(a)			(b)			(c)				(d)
Name	Init.	TriCycle Train	Variation	MAD Day	MAD Week	MAD Year	UST F1@0.5	UST P@0.5	UST R@0.5	UST AUPRC	Sensor Acc.
l3	L^3-Net	No	—	—	—	—	0.638	**0.767**	0.547	**0.751**	0.792
rand	Rand.	No	—	—	—	—	0.531	0.697	0.429	0.632	0.721
rand-tc	Rand.	Yes	—	0.480	0.508	0.562	0.622	0.734	0.540	0.712	0.781
l3-tc-llr	L^3-Net	Yes	Low LR	0.370	0.531	0.540	0.638	0.764	0.548	0.739	0.824
l3-tc-hlr	L^3-Net	Yes	High LR	**0.338**	0.443	0.545	0.638	0.749	0.556	0.737	**0.851**
rand-tc-rs	Rand.	Yes	Rand. Sampling	0.416	0.508	0.542	0.610	0.739	0.520	0.702	0.801
rand-tc-pcen	Rand.	Yes	PCEN	0.351	**0.423**	**0.444**	**0.650**	**0.767**	**0.564**	0.744	0.831

Table 1: A summary of the embedding models and their respective evaluation metrics on different tasks. (a) Description of the embedding model variations, including the weight initialization scheme, if TriCycle training was used, and any other miscellaneous details. (b) The pretext task results, evaluated using *median angular displacement* (MAD). A lower value of MAD indicates better performance on the temporal cycle prediction task. (c) The downstream urban sound tagging results, evaluated using micro-averaged F1-score, precision, and recall (using a threshold of 0.5), as well as micro-averaged AUPRC. (d) The downstream sensor classification results, evaluated using micro-averaged accuracy.

weight decay factor of 10^{-5}. Recording-level predictions are obtained by simply taking an arithmetic mean of the outputs of each frame. We evaluated each multi-class classification model on the test set recordings using multi-class accuracy.

5. RESULTS AND DISCUSSION

A summary of the results can be found in Table 1. For the urban sound tagging task, we find the baseline TriCycle model (*rand-tc*) performs better than the randomly initialized audio subnetwork when evaluated on AUPRC (0.71 vs 0.63). However, it does not outperform L^3-Net (*l3*) (0.75). But when we preprocess the input with PCEN, *rand-tc-pcen* performs comparably to *l3* on AUPRC (0.74 vs 0.75) and even slightly better when evaluated on F1-score at the 0.5 threshold (0.65 vs 0.64). When random sampling is used instead high-activity sampling in baseline (*rand-tc-rs*), we see a slight drop in performance. While L^3-Net trained only on AVC (*l3*) performs quite well, training the model further using TriCycle (i.e., *l3-tc-llr*, *l3-tc-hlr*) does not improve performance. In fact, it actually decreases slightly at both learning rates. When we compare the class-wise performance for *rand-tc-pcen* and *l3*, we find they perform similarly on all of the classes except for *non-machinery impact* and *powered saw*—*rand-tc-pcen* performs better on the former (0.42 vs 0.32 AUPRC) but worse on the latter (0.63 vs 0.73 AUPRC). For the sensor classification task, *rand-tc* performs the worst, followed by *l3*, and the TriCycle model initialized with L^3-Net and trained with a high-learning rate (*l3-tc-hlr*) performs the best.

These results indicate that while TriCycle alone can produce embeddings which perform better than those from a randomly initialized baseline, it requires help from other mechanisms—e.g., high-activity sampling and PCEN—to achieve performance comparable to L^3-Net on urban sound tagging. These mechanisms may help the model focus on foreground sound events rather than the acoustic background. The sensor classification results provide some evidence to this. With high-activity sampling enabled (*rand-tc*), UST AUPRC increases while sensor accuracy decreases—a sign that without high-activity sampling (*rand-tc-rs*), the model overfits to the acoustic background. With PCEN also enabled (*rand-tc-pcen*), UST AUPRC increases even more but sensor accuracy also increases—a sign not of overfitting to the acoustic background but rather that *rand-tc-pcen* may be learning location-dependent events.

While overall PCEN helps TriCycle, the poor results for the *powered saw* class highlight how it can harm performance too. Powered saws like walk-behind concrete saws produce loud, sustained sounds lacking in amplitude and frequency modulation, and therefore PCEN may treat such sounds as stationary background noise and effectively smooth them out. Since it is difficult to choose a single set of PCEN parameters that preserves and emphasizes a diverse set of sound classes, in future work we will investigate stacks of PCEN representations, configured with a range of time constants.

Our results also indicate that there is room to improve the pretext task, which may help learning the embedding. However, it does appear that PCEN processing generally improves performance on temporal cycle prediction, suggesting that foreground events are more indicative of cycle characteristics than background activity. In future work, we plan to investigate alternative loss formulations that restrict the output space to the unit circle and architecture changes which allow for groups of recordings to be trained simultaneously using late fusion. We believe the sound events present in a group of recordings from the same point in a cycle may more clearly indicate their cycle phase. The pretext task results also indicate that some cycles are easier to predict than others. Past self-supervision research indicates that the difficulty of the pretext task may affect downstream performance [8]; therefore, we also plan to investigate the benefit of each cycle prediction task in future work.

6. CONCLUSION

TriCycle is a self-supervised audio embedding model trained by predicting the time of recording, encoded as phase within day, week, and year cycles. To our knowledge, this is the first self-supervised embedding model trained on long-term temporal structure. The goal of this work is to develop a model for training dataset-specific embeddings with single-modal data that *outperform* general-domain embeddings. While we have not quite achieved that goal in this paper, we have validated our approach on an urban sound tagging task, *matching* performance of a state-of-the-art embedding trained on multi-modal data. In our analysis, we found that it is imperative to focus the model on sound events of interest for good performance, and we have outlined several future steps that may further improve the model. While we have tested this novel model on audio, it may also be well-suited for datasets from other sensor networks also having dense, longitudinal, timestamped data and may enable training embeddings previously not possible.

7. REFERENCES

[1] J. P. Bello, C. Silva, O. Nov, R. L. Dubois, A. Arora, J. Salamon, C. Mydlarz, and H. Doraiswamy, "Sonyc: A system for monitoring, analyzing, and mitigating urban noise pollution," *Commun. ACM*, vol. 62, no. 2, pp. 68–77, Jan. 2019.

[2] V. R. de Sa, "Learning classification with unlabeled data," in *Advances in neural information processing systems*, 1994, pp. 112–119.

[3] 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*, 2015, pp. 1422–1430.

[4] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[5] D. Wei, J. J. Lim, A. Zisserman, and W. T. Freeman, "Learning and using the arrow of time," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8052–8060.

[6] X. Wang, A. Jabri, and A. A. Efros, "Learning correspondence from the cycle-consistency of time," in *CVPR*, 2019.

[7] I. Misra, C. L. Zitnick, and M. Hebert, "Shuffle and learn: unsupervised learning using temporal order verification," in *European Conference on Computer Vision*. Springer, 2016, pp. 527–544.

[8] B. Fernando, H. Bilen, E. Gavves, and S. Gould, "Self-supervised video representation learning with odd-one-out networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3636–3645.

[9] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba, "The sound of pixels," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 570–586.

[10] 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.

[11] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba, "Ambient sound provides supervision for visual learning," in *European conference on computer vision*. Springer, 2016, pp. 801–816.

[12] B. Korbar, D. Tran, and L. Torresani, "Cooperative learning of audio and video models from self-supervised synchronization," in *Advances in Neural Information Processing Systems*, 2018, pp. 7763–7774.

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

[14] R. Arandjelovi and A. Zisserman, "Look, listen and learn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[15] R. Arandjelovic and A. Zisserman, "Objects that sound," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 435–451.

[16] 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.

[17] N. Turpault, R. Serizel, and E. Vincent, "Semi-supervised triplet loss based learning of ambient audio embeddings," in *ICASSP*, 2019.

[18] Y. F. Phillips, M. Towsey, and P. Roe, "Revealing the ecological content of long-duration audio-recordings of the environment through clustering and visualisation," *PloS one*, vol. 13, no. 3, p. e0193345, 2018.

[19] M. Batty, "Big data, smart cities and city planning," *Dialogues in Human Geography*, vol. 3, no. 3, pp. 274–279, 2013, pMID: 29472982.

[20] M. Cartwright, A. E. M. Mendez, G. Dove, J. Cramer, V. Lostanlen, H.-H. Wu, J. Salamon, O. Nov, and J. P. Bello, "Sonyc urban sound tagging (sonyc-ust): a multilabel dataset from an urban acoustic sensor network," Mar 2019. [Online]. Available: https://zenodo.org/record/2590742

[21] R. Simpson, K. R. Page, and D. De Roure, "Zooniverse: Observing the world's largest citizen science platform," in *Proceedings of the 23rd International Conference on World Wide Web*, ser. WWW '14 Companion. New York, NY, USA: ACM, 2014, pp. 1049–1054.

[22] M. Cartwright, G. Dove, A. E. Méndez Méndez, J. P. Bello, and O. Nov, "Crowdsourcing multi-label audio annotation tasks with citizen scientists," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19. New York, NY, USA: ACM, 2019, pp. 292:1–292:11.

[23] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ser. ICML'15. JMLR.org, 2015, pp. 448–456.

[24] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010, pp. 807–814.

[25] Y. Wang, P. Getreuer, T. Hughes, R. F. Lyon, and R. A. Saurous, "Trainable frontend for robust and far-field keyword spotting," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5670–5674.

[26] V. Lostanlen, J. Salamon, M. Cartwright, B. McFee, A. Farnsworth, S. Kelling, and J. P. Bello, "Per-channel energy normalization: Why and how," *IEEE Signal Processing Letters*, vol. 26, no. 1, pp. 39–43, 2019.

[27] B. McFee, J. Salamon, and J. P. Bello, "Adaptive pooling operators for weakly labeled sound event detection," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 26, no. 11, pp. 2180–2193, 2018.