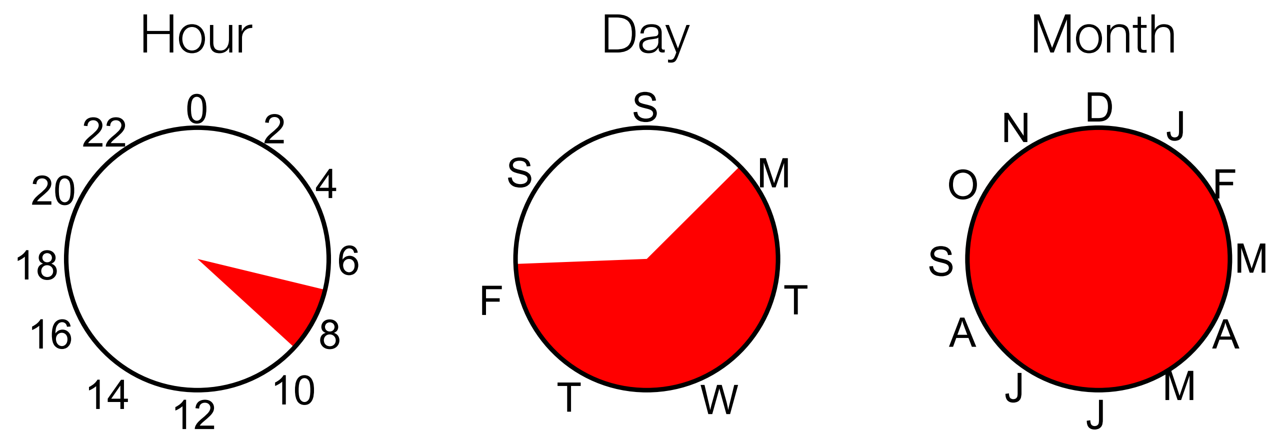
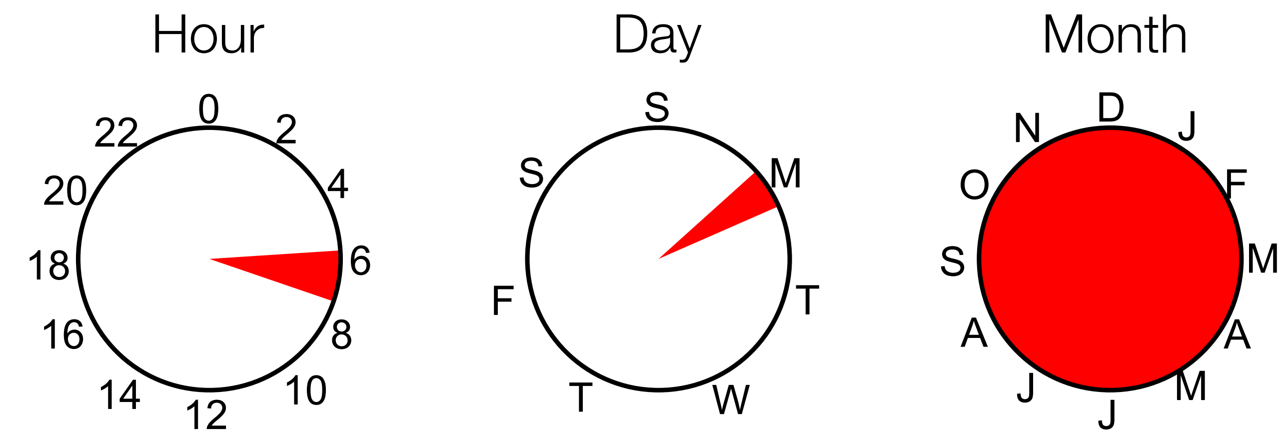
TriCycle: Audio Representation Learning from Sensor Network Data Using Self-Supervision

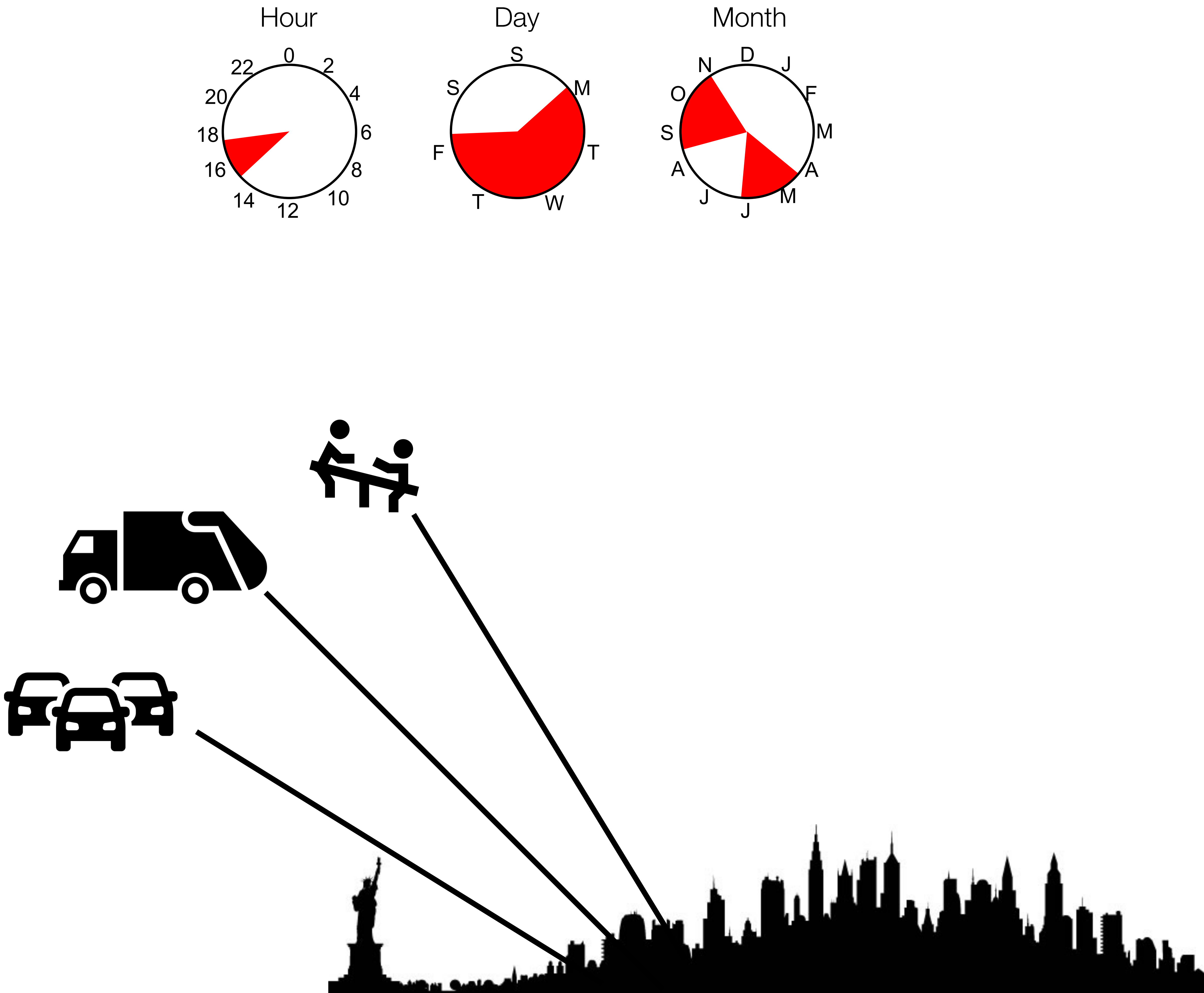
Mark Cartwright[1], Jason Cramer[1], Justin Salamon[2], Juan Pablo Bello[1]

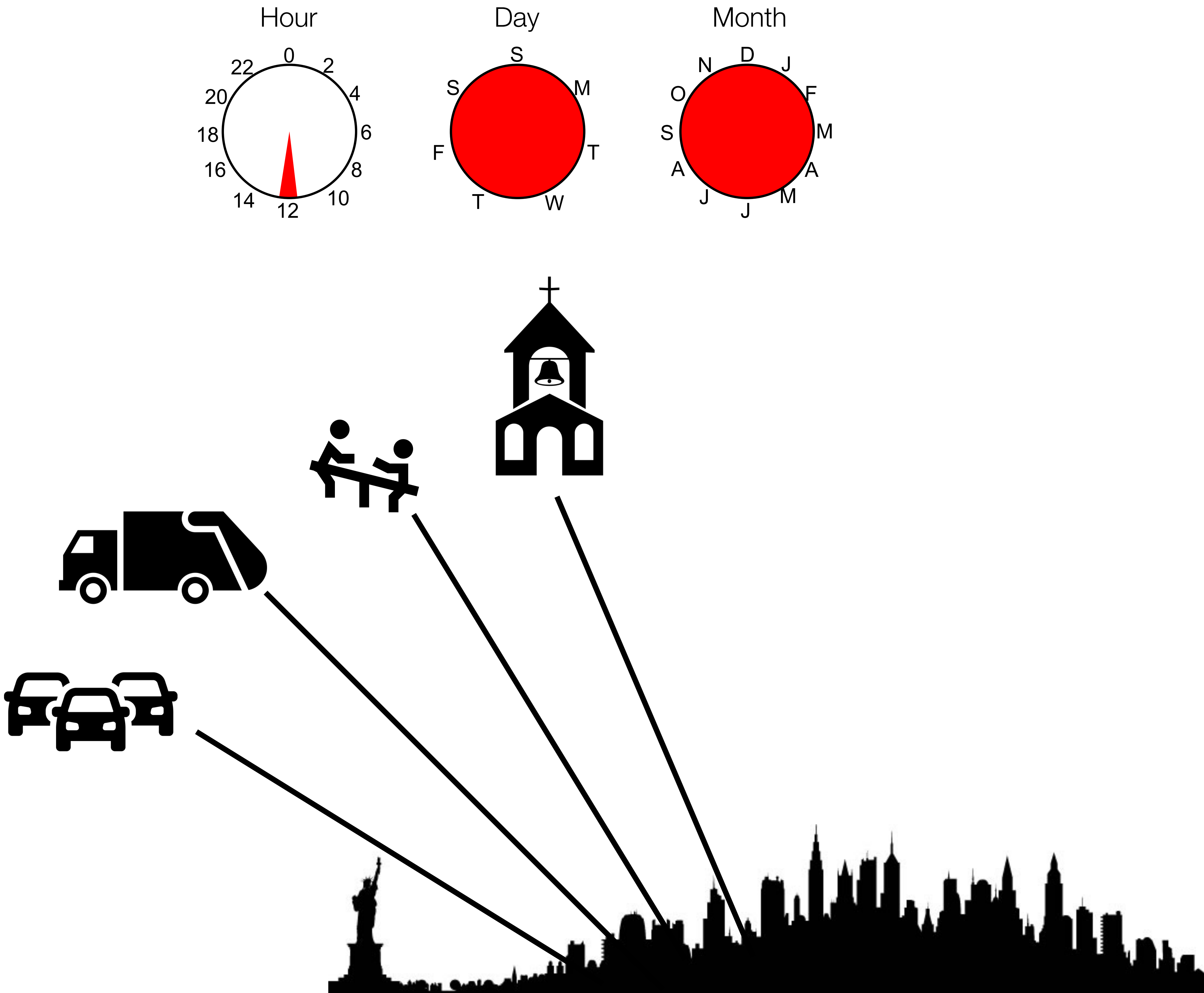
1. New York University Music and Audio Research Lab
2. Adobe Research

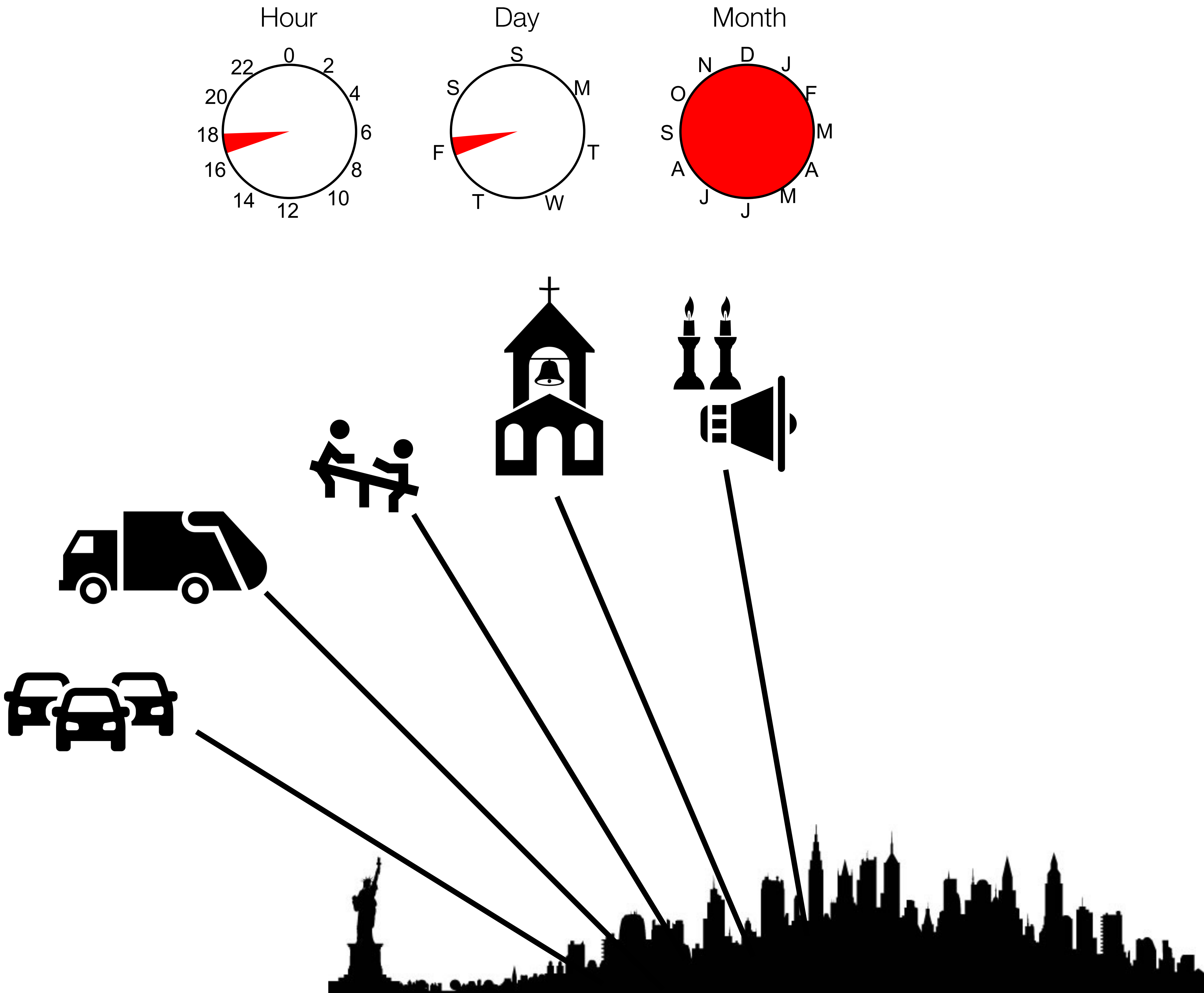

H&M
H&M


Hour
0
2
4
6
8
10
12
14
16
18
20
22
Day
S
M
T
W
T
F
S
Month
D
J
F
M
A
M
J
J
A
S
O
N

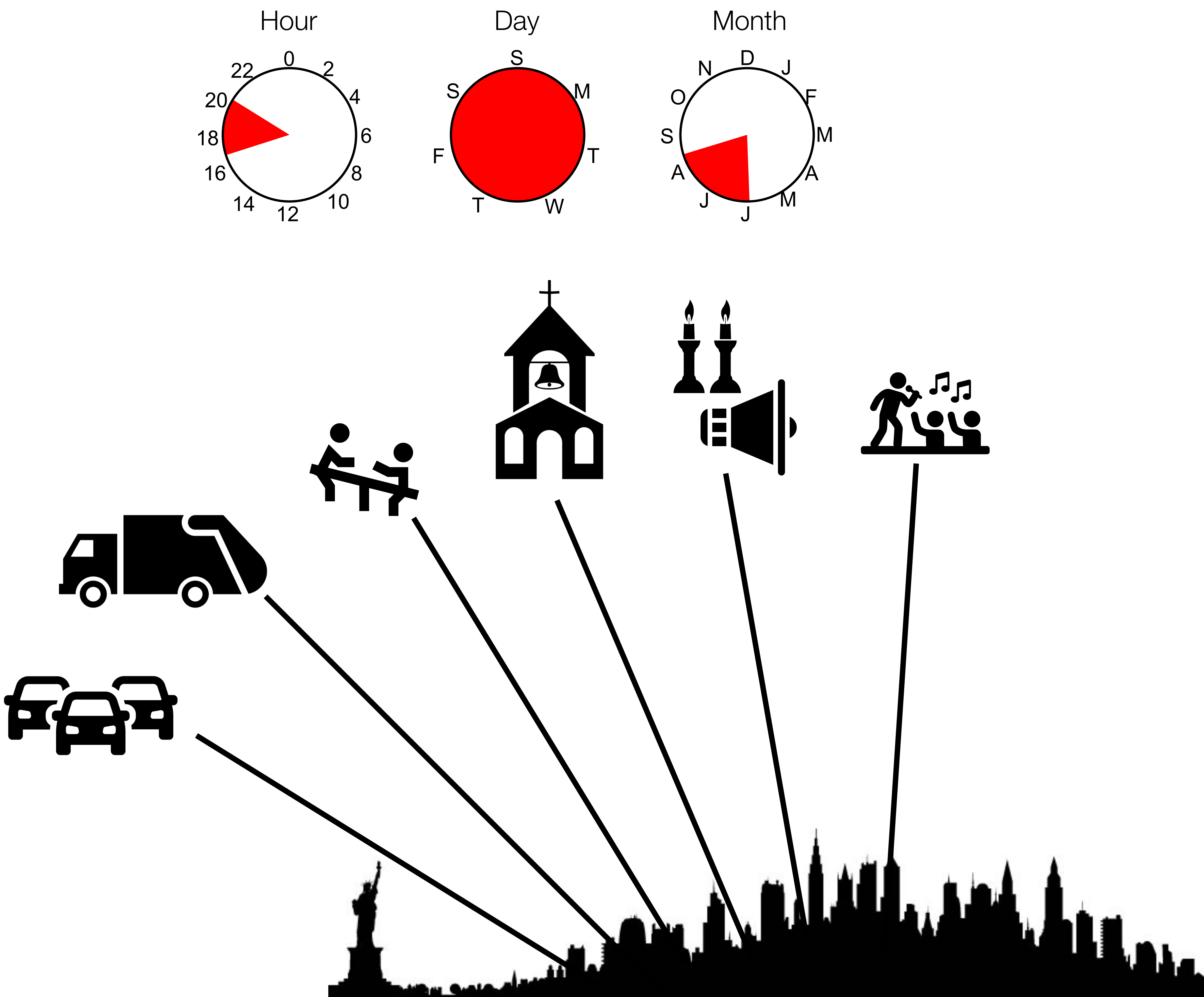

Hour
0
2
4
6
8
10
12
14
16
18
20
22
Day
S
M
T
W
T
F
S
Month
D
J
F
M
A
M
J
J
A
S
O
N


Hour
0
2
4
6
8
10
12
14
16
18
20
22
Day
S
M
T
W
T
F
S
Month
D
J
F
M
A
M
J
J
A
S
O
N

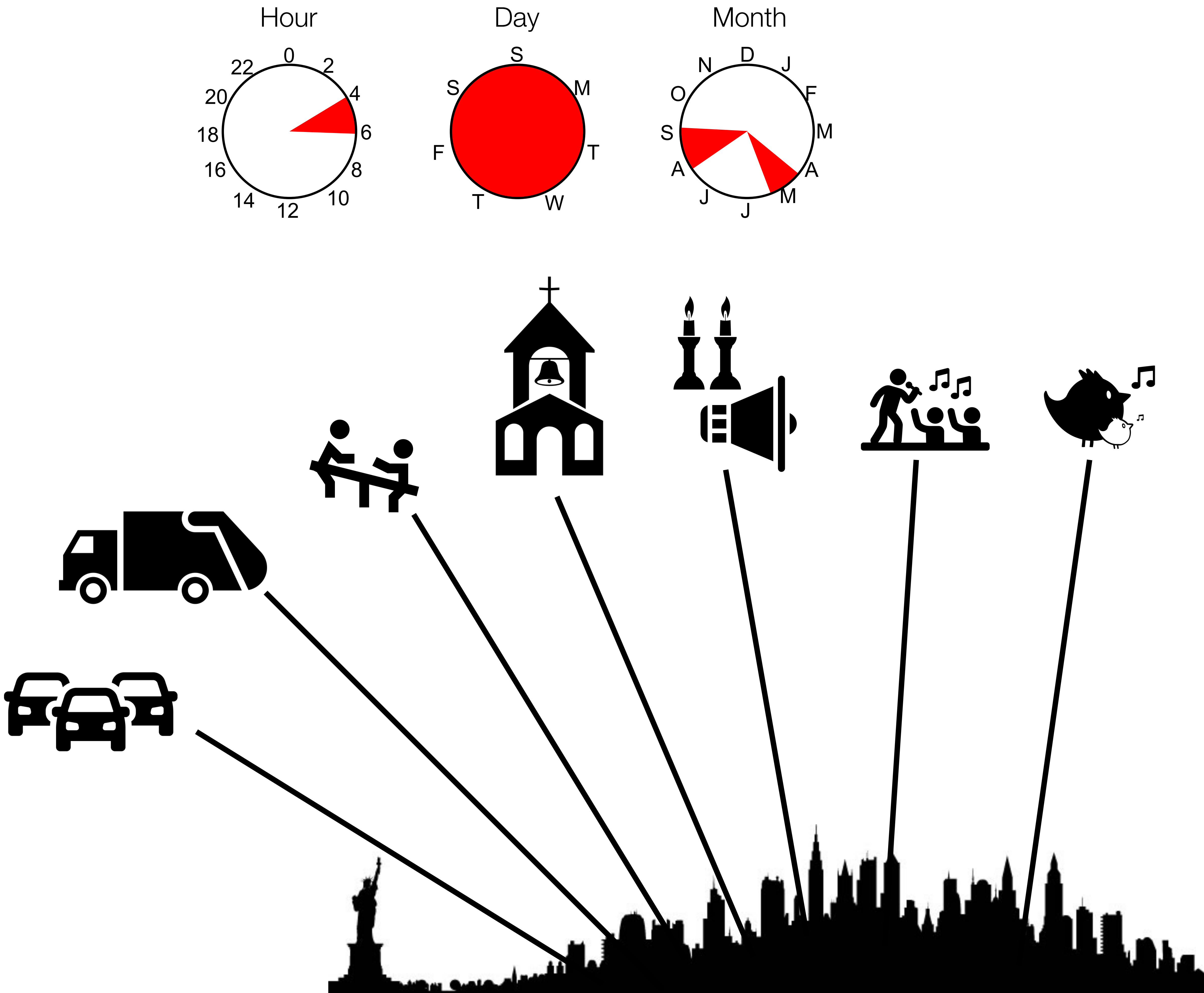

Hour
0
2
4
6
8
10
12
14
16
18
20
22
Day
S
M
T
W
T
F
S
Month
D
J
F
M
A
M
J
J
A
S
O
N



Hour
0
2
4
6
8
10
12
14
16
18
20
22
Day
S
M
T
W
T
F
S
Month
D
J
F
M
A
M
J
J
A
S
O
N



Hour
0
2
4
6
8
10
12
14
16
18
20
22
Day
S
M
T
W
T
F
S
Month
D
J
F
M
A
M
J
J
A
S
O
N



Hour
0
2
4
6
8
10
12
14
16
18
20
22
Day
S
M
T
W
T
F
S
Month
D
J
F
M
A
M
J
J
A
S
O
N

57

Fixed-location Sensors

130M

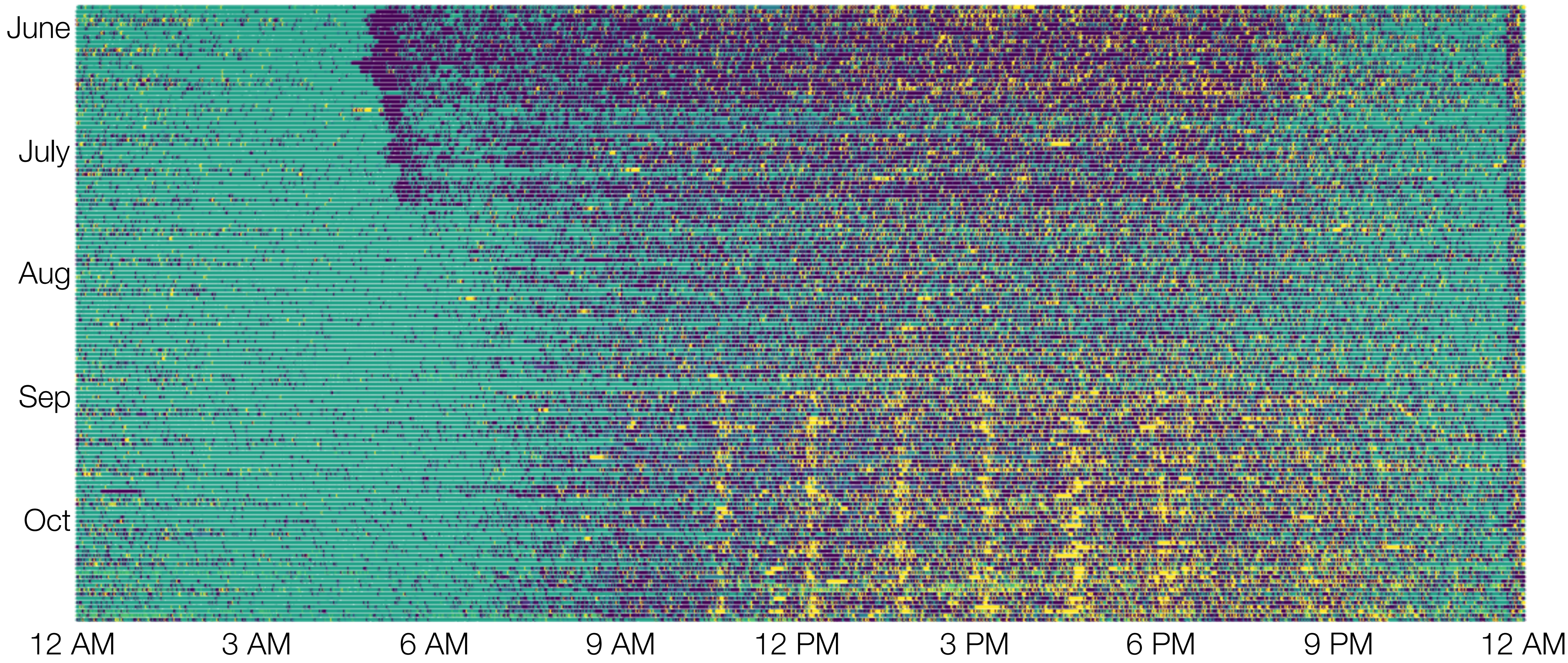
10 sec Recordings

40

Sensor Years of Audio

Long-term temporal structure in SONYC recordings

Predominant cluster (N=8) over 4 months for 1 sensor



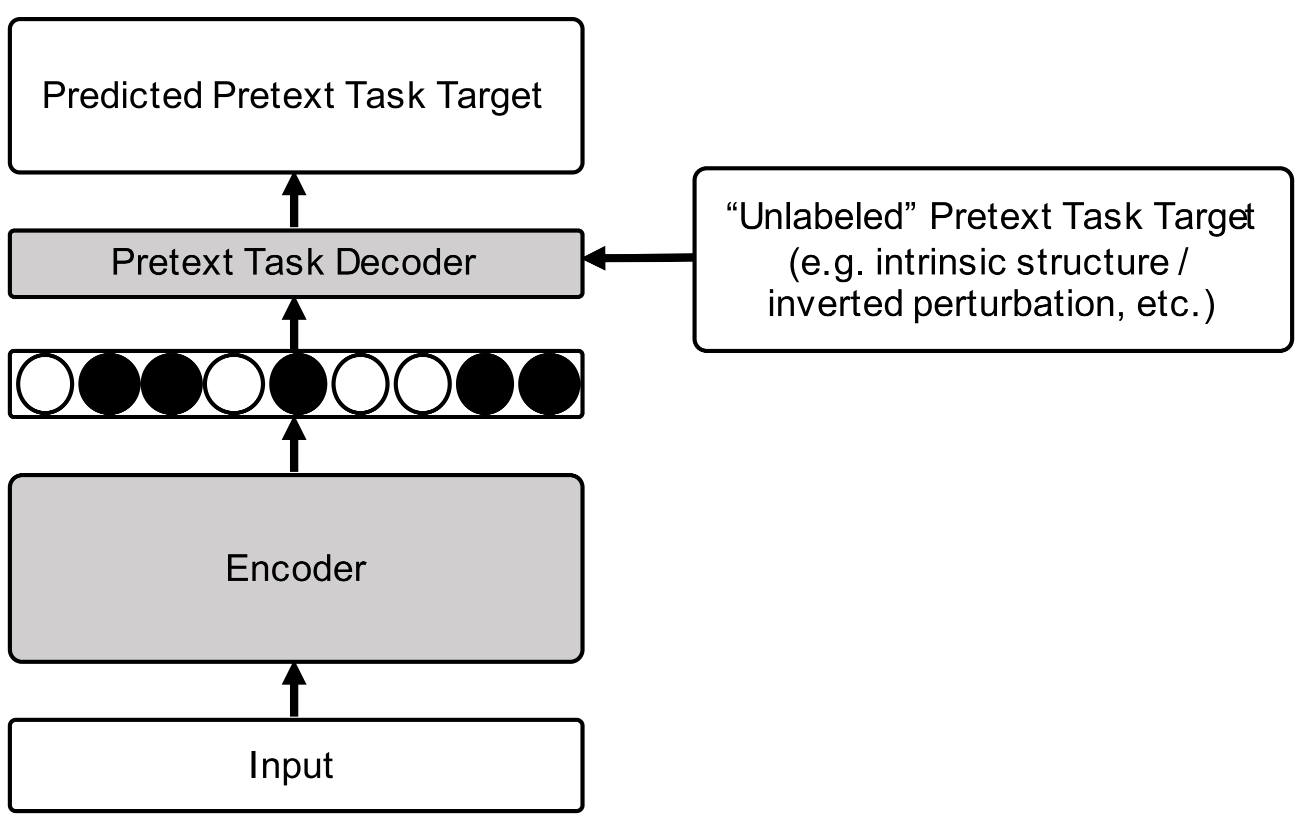
Long-term temporal structure in SONYC recordings

Cluster frequency grouped by hour of the day on SONYC recordings

Can we exploit this long-term seasonal structure for self-supervised audio representation learning?

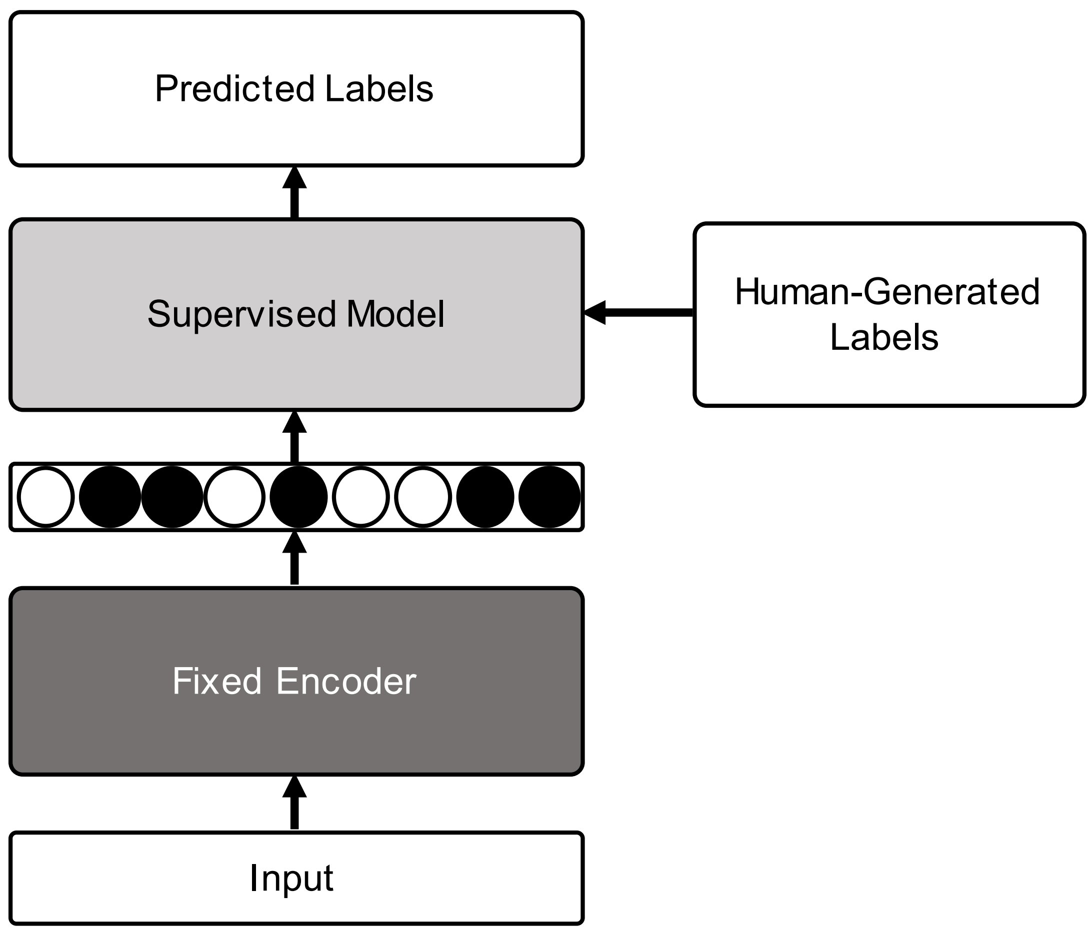
Self-supervised pretext task

- Learn representations (embeddings) by solving pretext tasks
- Pretext tasks exploit known intrinsic structure or estimate / invert a controlled perturbation
- Key is that pretext tasks **do not** require (human-generated) labels and are trained on **lots** of this "unlabeled" data

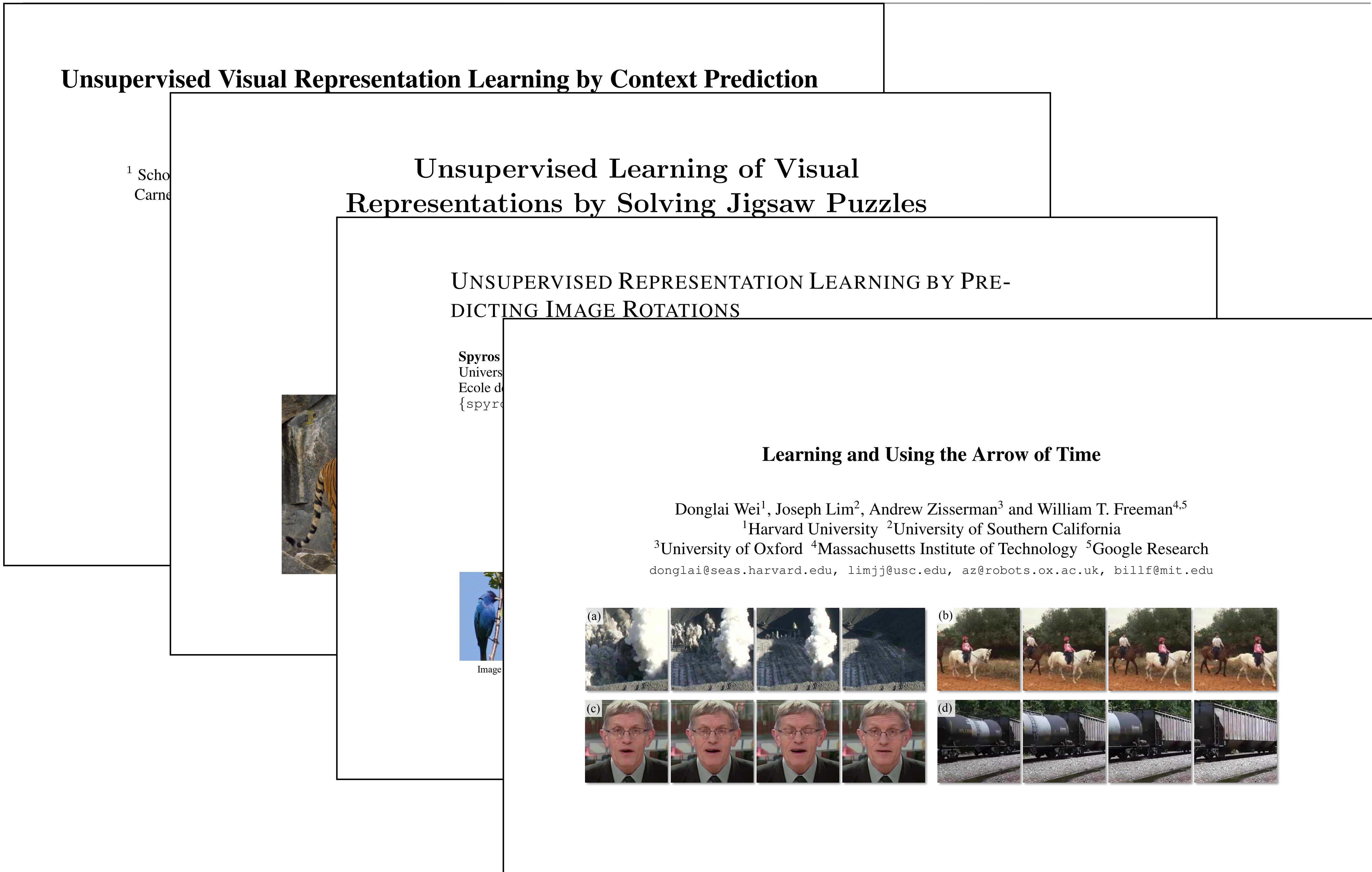


Supervised downstream task

- With learned representation as input, use **simpler**, **smaller** capacity supervised model with **fewer labeled examples** in a downstream task



Examples in computer vision


Unsupervised Visual Representation Learning by Context Prediction

Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles

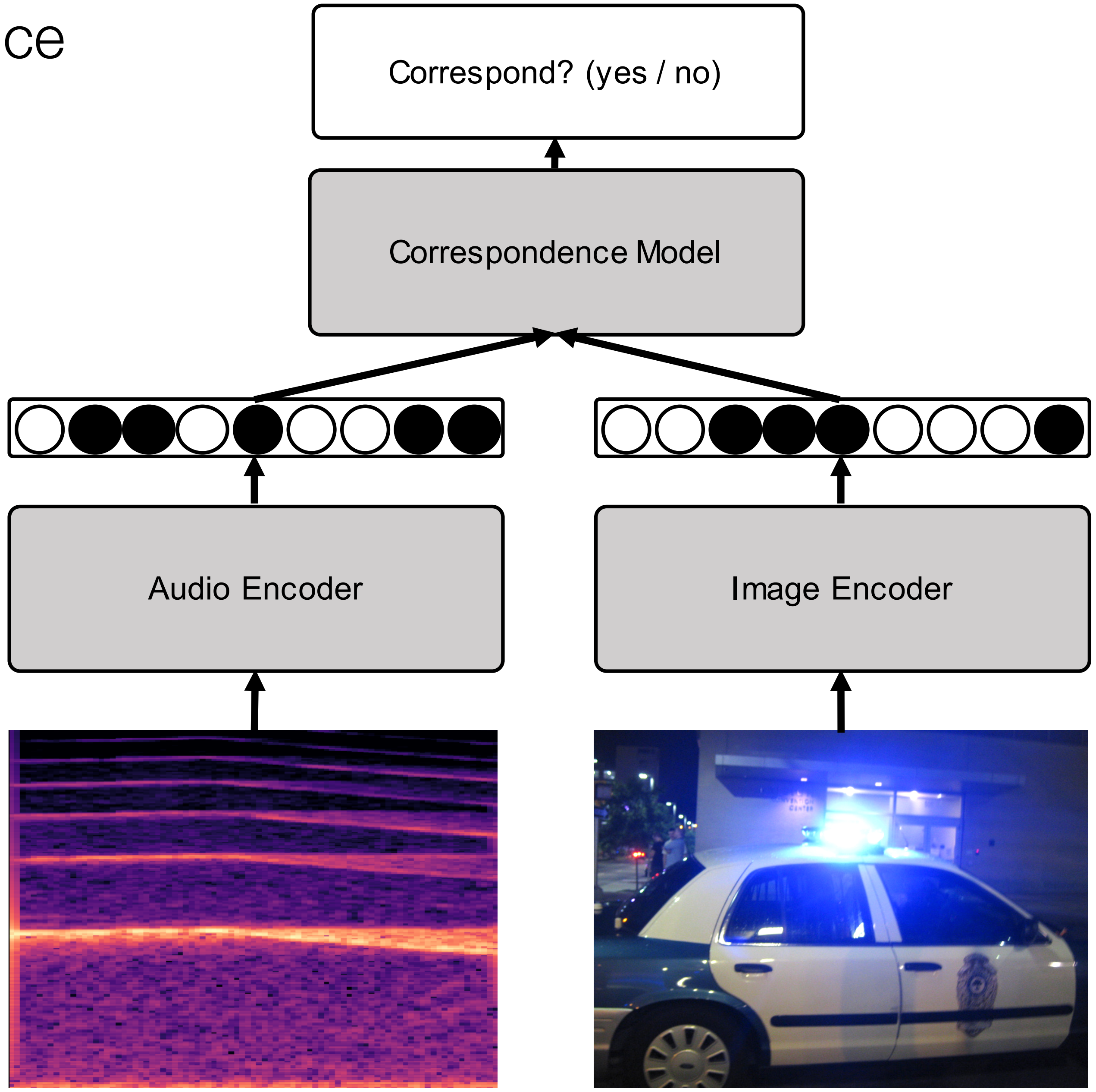
UNSUPERVISED REPRESENTATION LEARNING BY PREDICTING IMAGE ROTATIONS

Learning and Using the Arrow of Time

Donglai Wei[1], Joseph Lim[2], Andrew Zisserman[3] and William T. Freeman[4,5]
[1]Harvard University [2]University of Southern California
[3]University of Oxford [4]Massachusetts Institute of Technology [5]Google Research
donglai@seas.harvard.edu, limjj@usc.edu, az@robots.ox.ac.uk, billf@mit.edu

Examples in machine listening

- Arandjelovic & Zisserman, “Look, listen and learn” (L3), ICCV 2017
- Exploits audio-visual correspondence

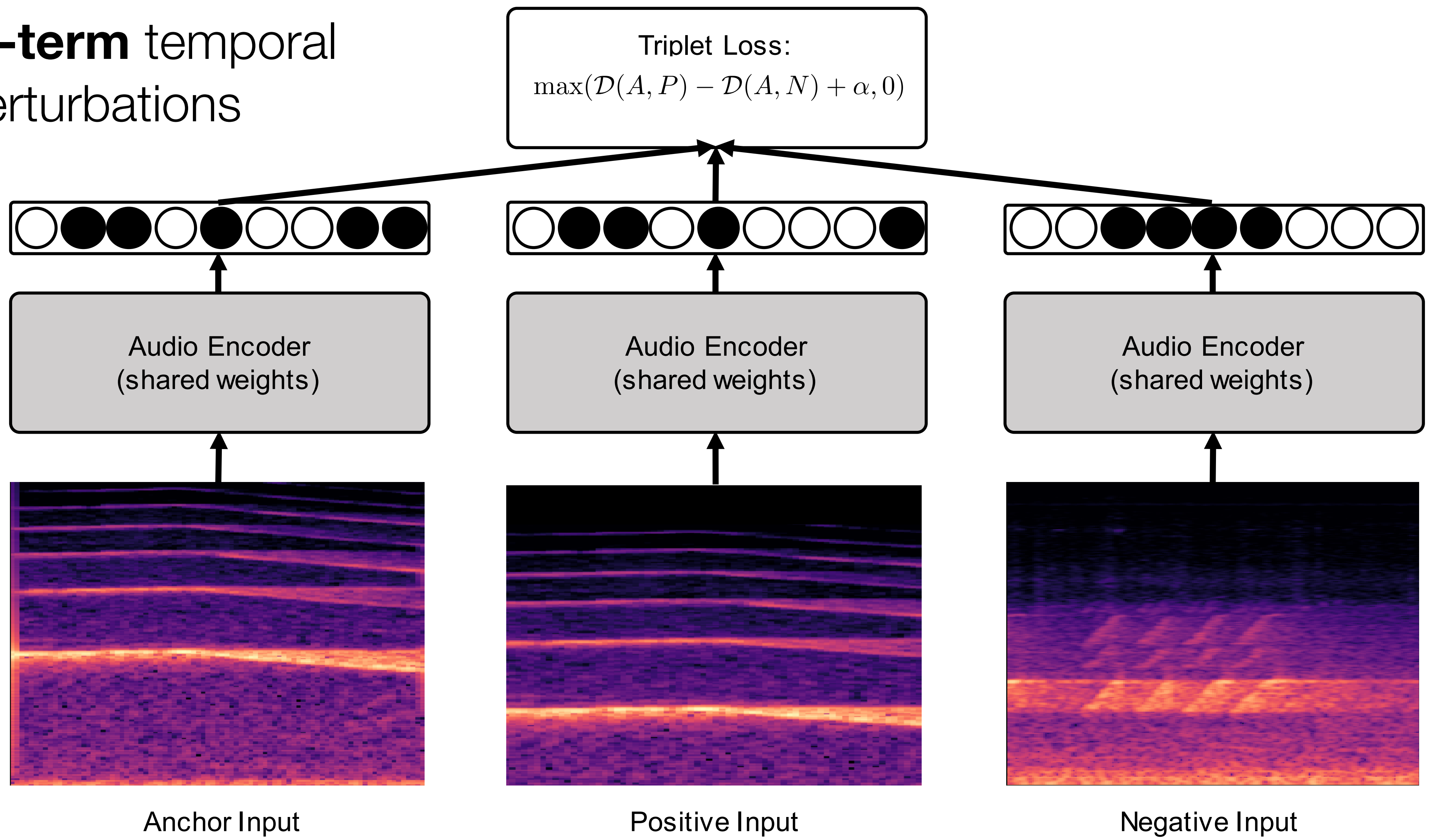


1 s Mel-spectrogram Input

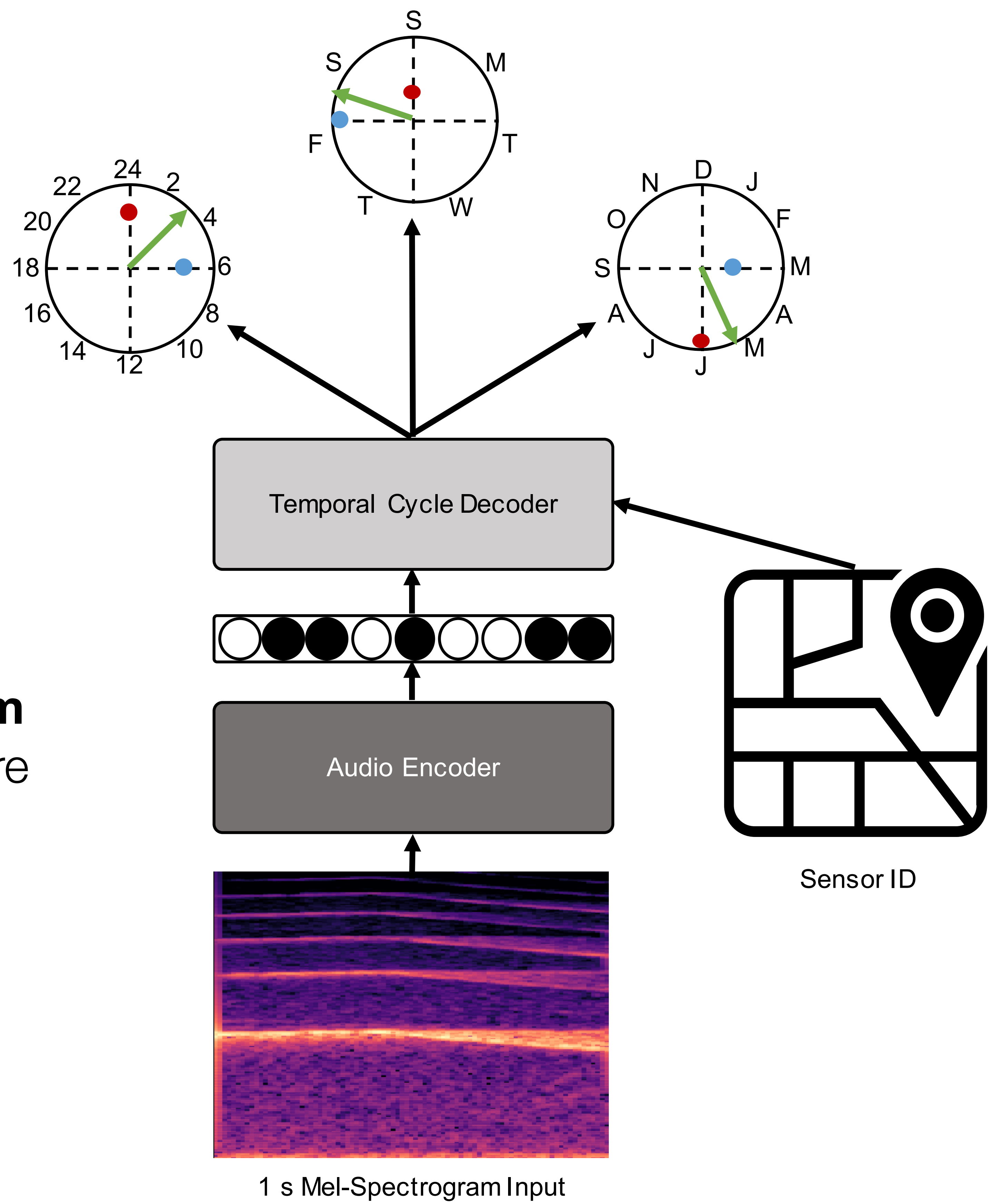
Single image video frame

Examples in machine listening

- Jansen, et al. “Unsupervised learning of semantic audio representations”, ICASSP 2018
- Exploits **short-term** temporal structure or perturbations

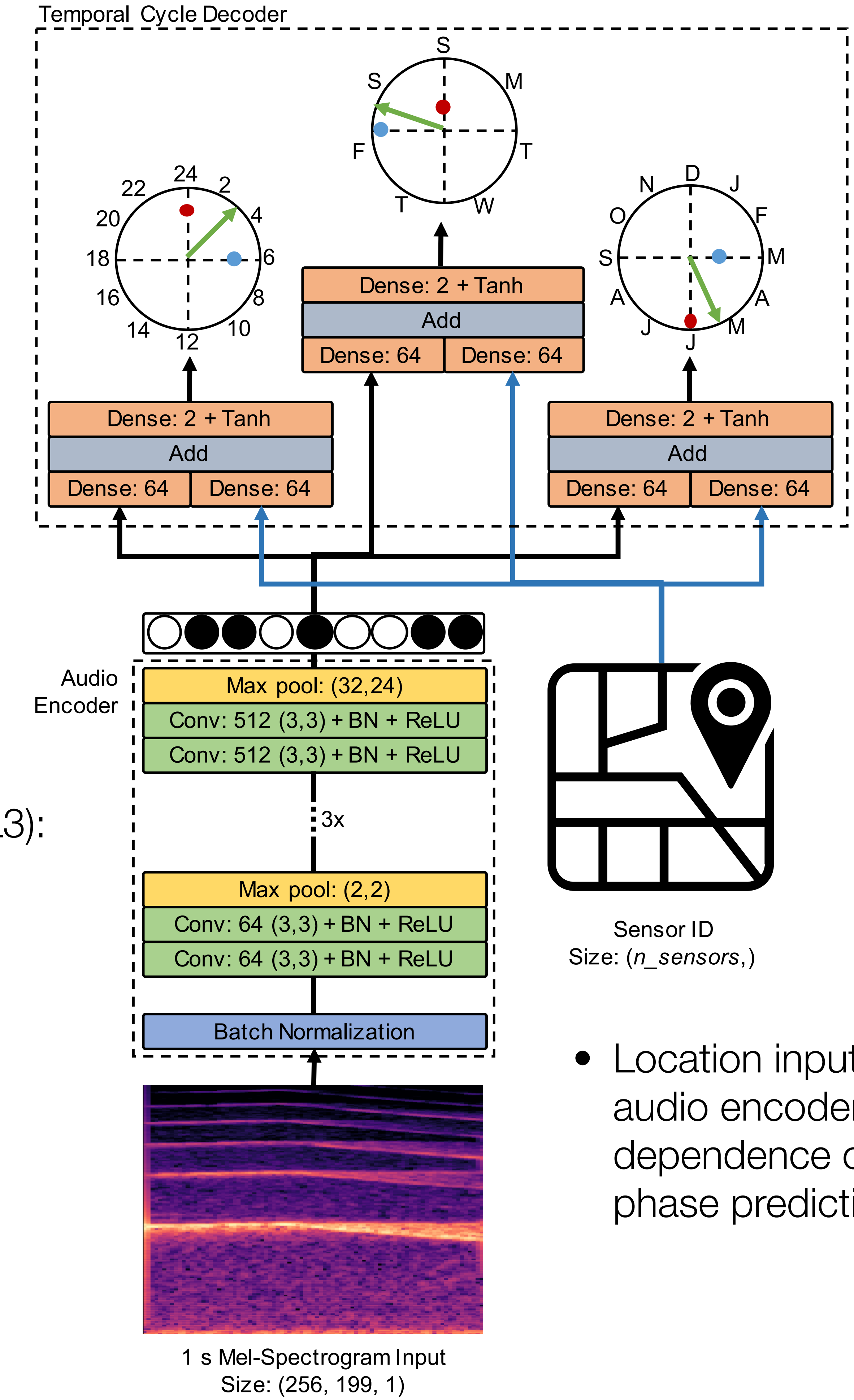


TriCycle Model



- We propose to exploit **long-term** temporal structure

TriCycle Model



- To avoid issues with phrase wrapping, phase encoded as $[\cos(\phi), \sin(\phi)]$ optimized with MSE loss

- Audio encoder same as “Look, Listen, and Learn” (L3):
  - Simple CNN
  - 4 convolutional blocks
  - Each with 2 conv. layers
    + max pooling

- Input:
  48kHz
  256-bin Mel spectrogram
  log-scaled magnitude
  5 ms hop

- Location input incorporated after audio encoder to account for location dependence of sound events in phase prediction

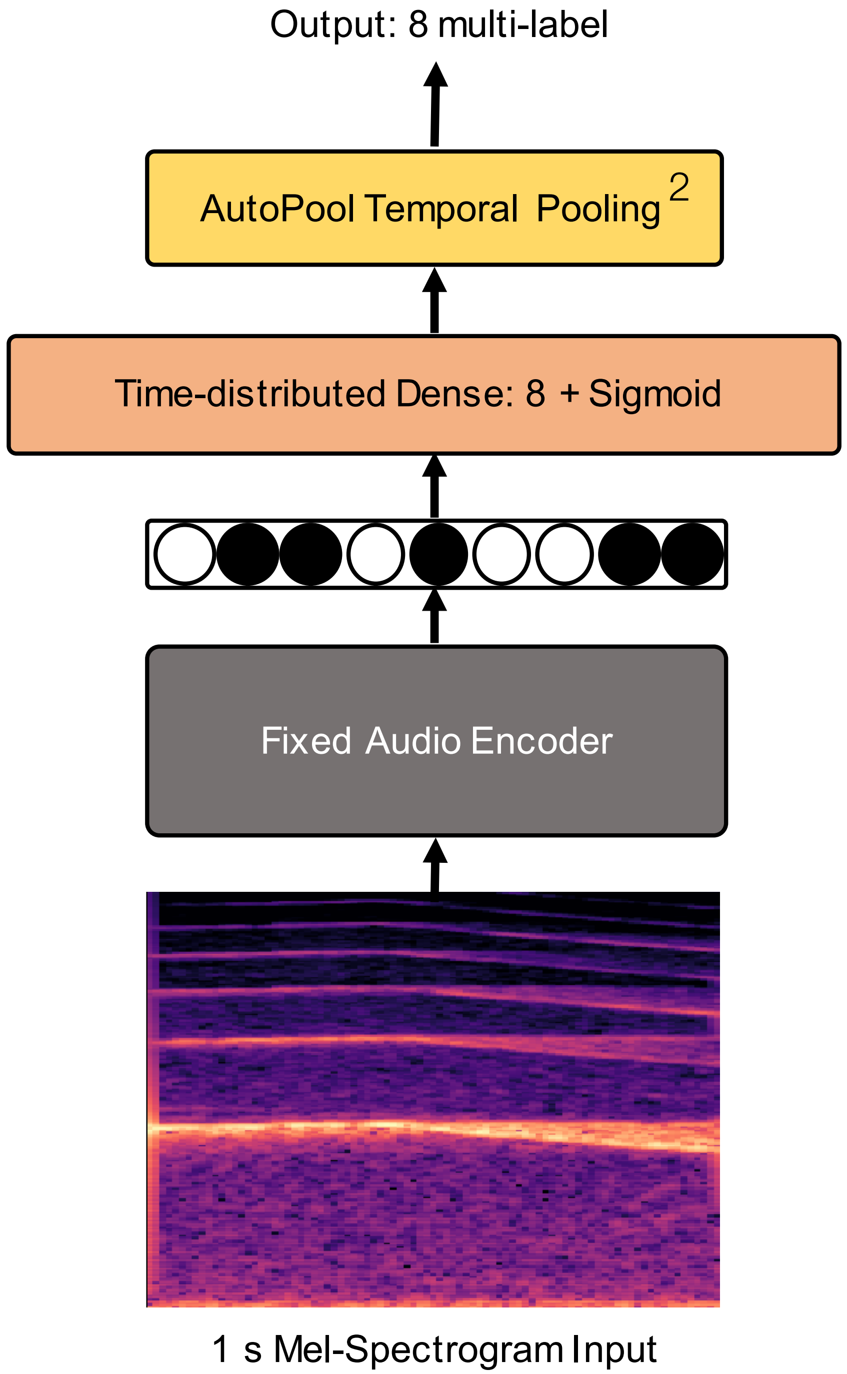
TriCycle Training

- Because of resource constraints, limited SONYC dataset to 2017 data from 25 sensors ~25M 10 sec recordings (69k hours)
- Randomly sampled
- 1500 “epochs” (24M training examples)

Supervised downstream task: Urban sound tagging

SONYC Urban Sound Tagging (UST) Dataset[1]

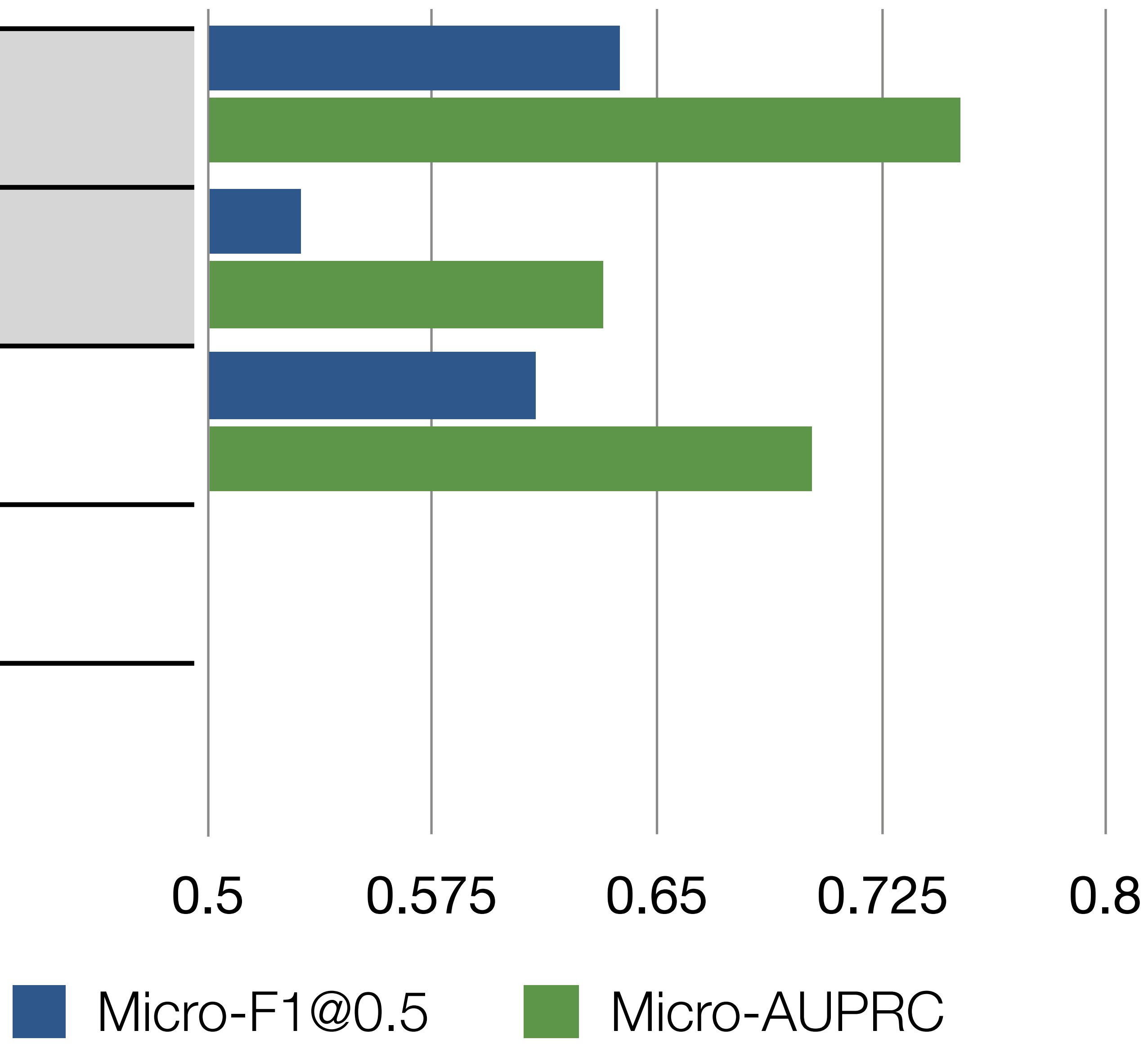
- labeled subset of SONYC data
- v0.1 Released in March
- 2019 DCASE Urban Sound Tagging Challenge dataset
- 10 sec recordings from SONYC sensors
  2351 training
  443 validation
  274 test (*did not use*)
- Weak multi-label annotation on 23 fine-level classes from 8 coarse-level groups (we used the coarse labels): *engine, machinery impact, non-machinery impact, powered saw, alert signal, music, human voice, dog*
- 3 Zooniverse volunteer annotators per recording
  Used minority vote to aggregate
- Validation and test set annotated by SONYC team



1. Cartwright, et al. “SONYC Urban Sound Tagging (SONYC-UST): A multilabel dataset from an urban acoustic sensor network”, DCASE 2019
2. McFee, Salamon, Bello. “Adaptive pooling operators for weakly labeled sound event detection”, TASLP 2018

Urban sound tagging results with TriCycle

Initialization	TriCycle training
L3	No
Random	No
Random	Yes



Strategies to focus on foreground events: High-activity sampling

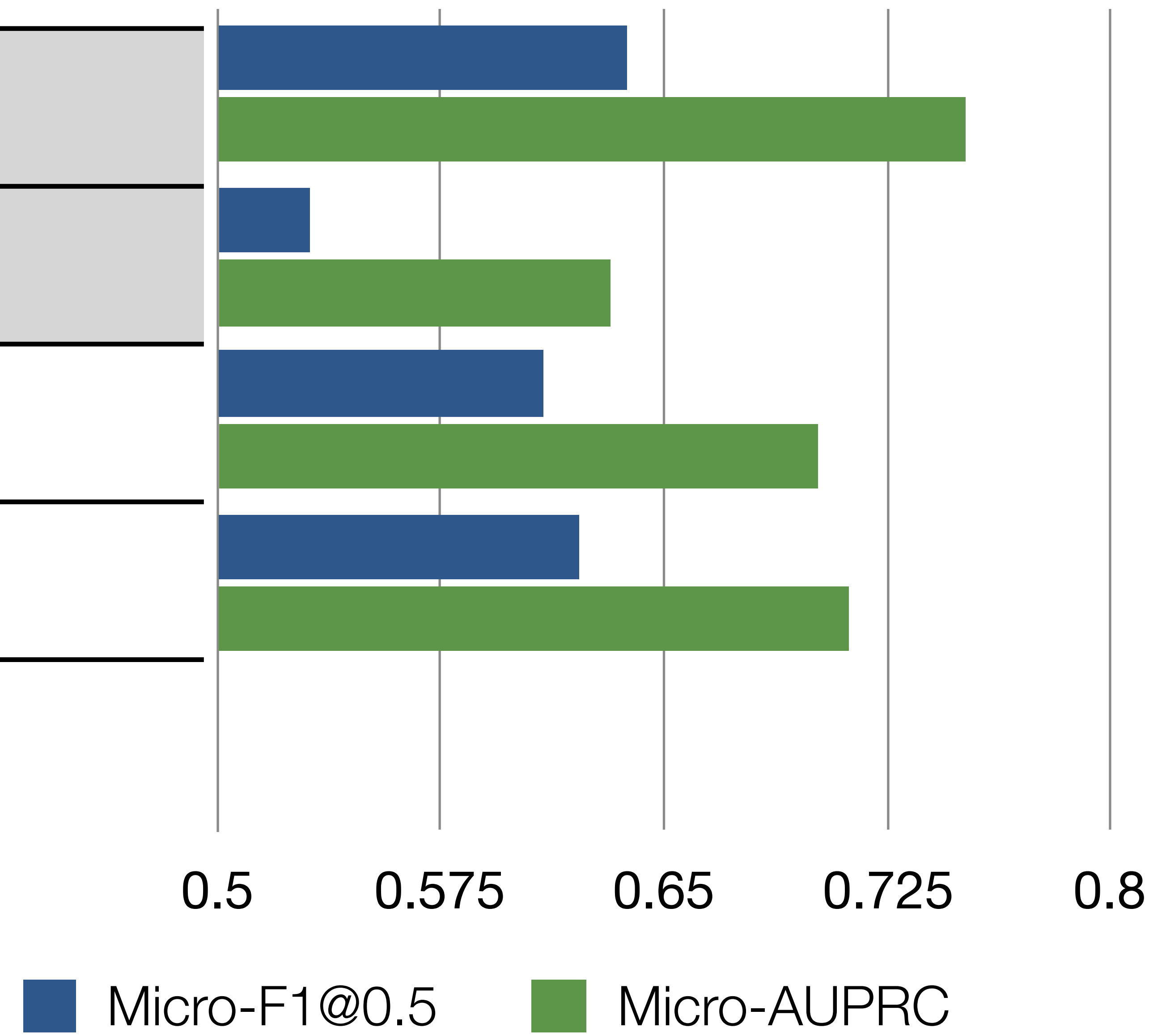
- Focus on high activity regions but still evenly sample each hour
- Compute SPL “activity” metric for each 10 s recording (SPL b/c precomputed):

$$\sqrt{\sum_{n=0}^{79}(d_{m,n} - d_{m,n-1})^2}$$

  for SPL sequence d of length 80 (i.e., 10 s with 0.125 s step size) from sensor m
- Only sample from top 15 percent of each hour
- Within each 10 s recording, sample 1 s clip, weighting by SPL

Urban sound tagging results with TriCycle

Initialization	TriCycle training	High-activity sampling
L3	No	
Random	No	
Random	Yes	
Random	Yes	Yes



Focusing on foreground events: Per-Channel Energy Normalization (PCEN)

Pre-process with Per-Channel Energy Normalization (PCEN)[1]

- Spectrogram processing that Gaussianizes and decorrelates frequency channels while retaining sound events of interest (parameter hand tuned based on recommendations in [2])

$$\mathbf{PCEN}(t,f) = \left(\frac{\mathbf{E}(t,f)}{\left(\varepsilon + (\mathbf{E} \overset{t}{\ast} \phi_T)(t,f)\right)^{\alpha}} + \delta \right)^{r} - \delta^{r}$$

Temporal integration

Automatic gain control

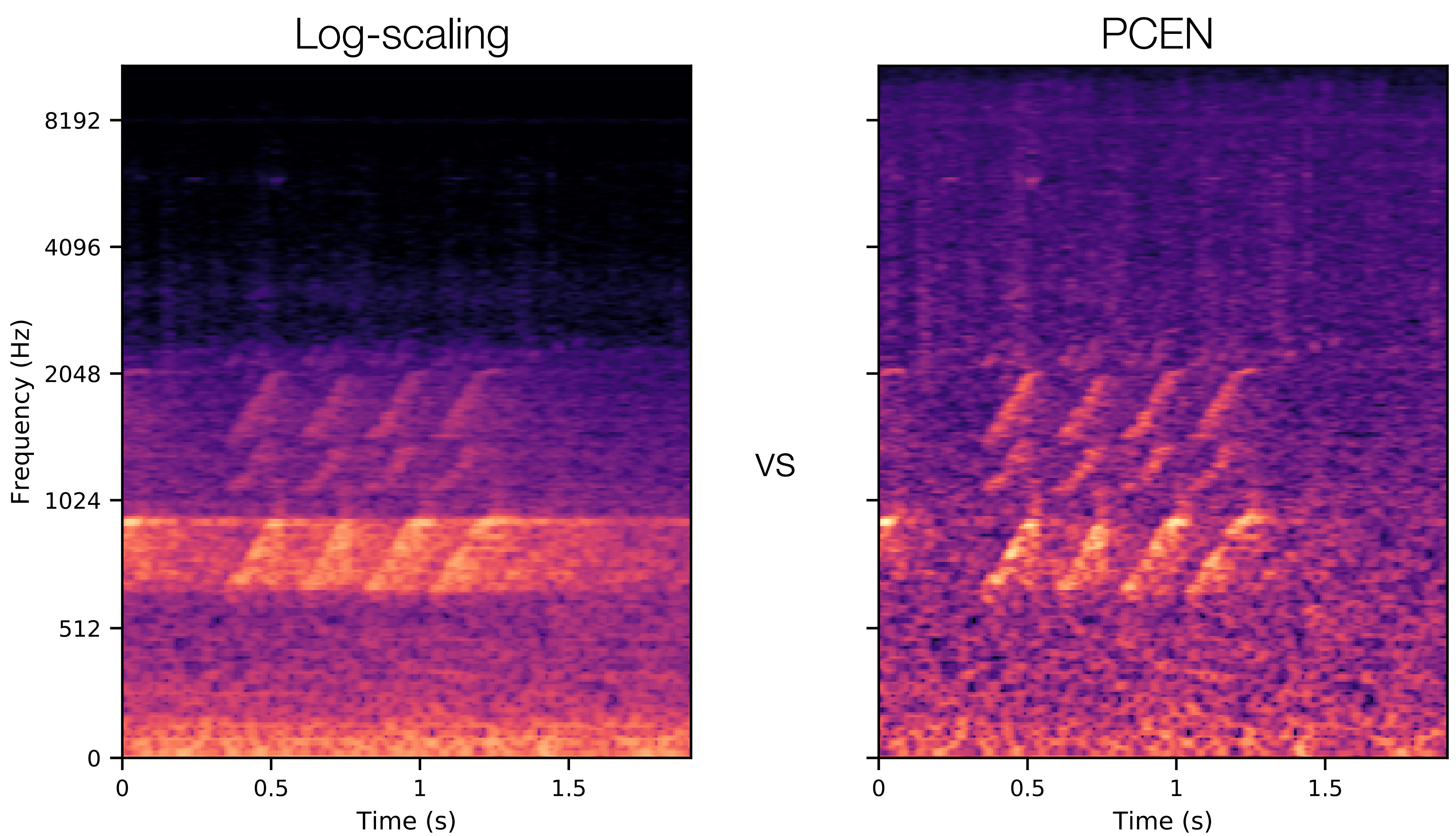
Dynamic range compression

1. Wang, et al. “Trainable frontend for robust and far-field keyword spotting”, ICASSP 2017
2. Lostanlen, Salamon, Cartwright, McFee, Farnsworth, Kelling, Bello, “Per-Channel Energy Normalization: Why and How”, SPL 2019

Strategies to focus on foreground events: Per-Channel Energy Normalization (PCEN)

Pre-process with Per-Channel Energy Normalization (PCEN)[1]

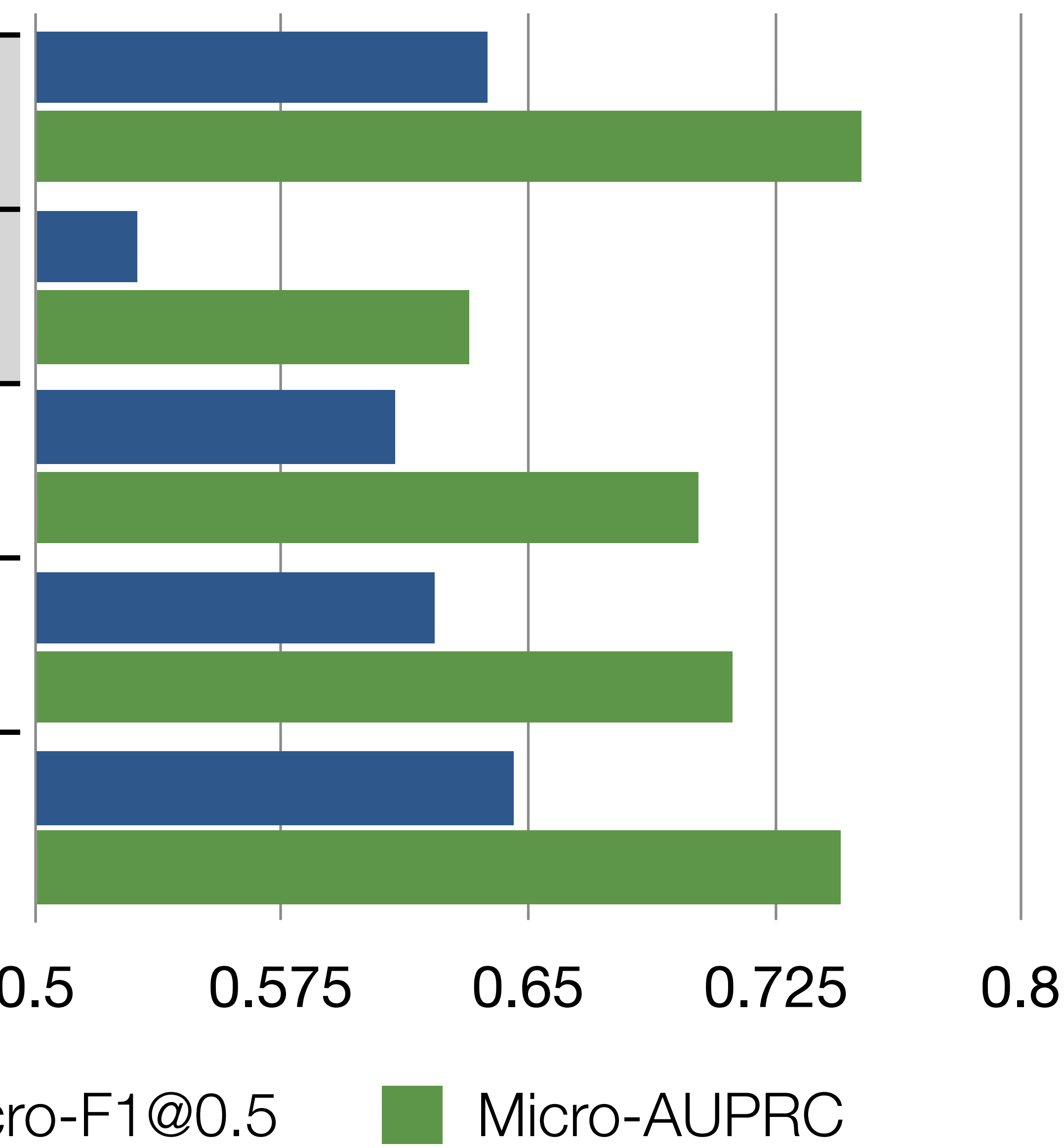
- Spectrogram processing that Gaussianizes and decorrelates frequency channels while retaining sound events of interest (parameter hand tuned based on recommendations in [2])



1. Wang, et al. “Trainable frontend for robust and far-field keyword spotting”, ICASSP 2017
2. Lostanlen, Salamon, Cartwright, McFee, Farnsworth, Kelling, Bello, “Per-Channel Energy Normalization: Why and How”, SPL 2019

Urban sound tagging results with TriCycle

Initialization	TriCycle training	High-activity sampling	PCEN
L3	No		
Random	No		
Random	Yes		
Random	Yes	Yes	
Random	Yes	Yes	Yes



Future work

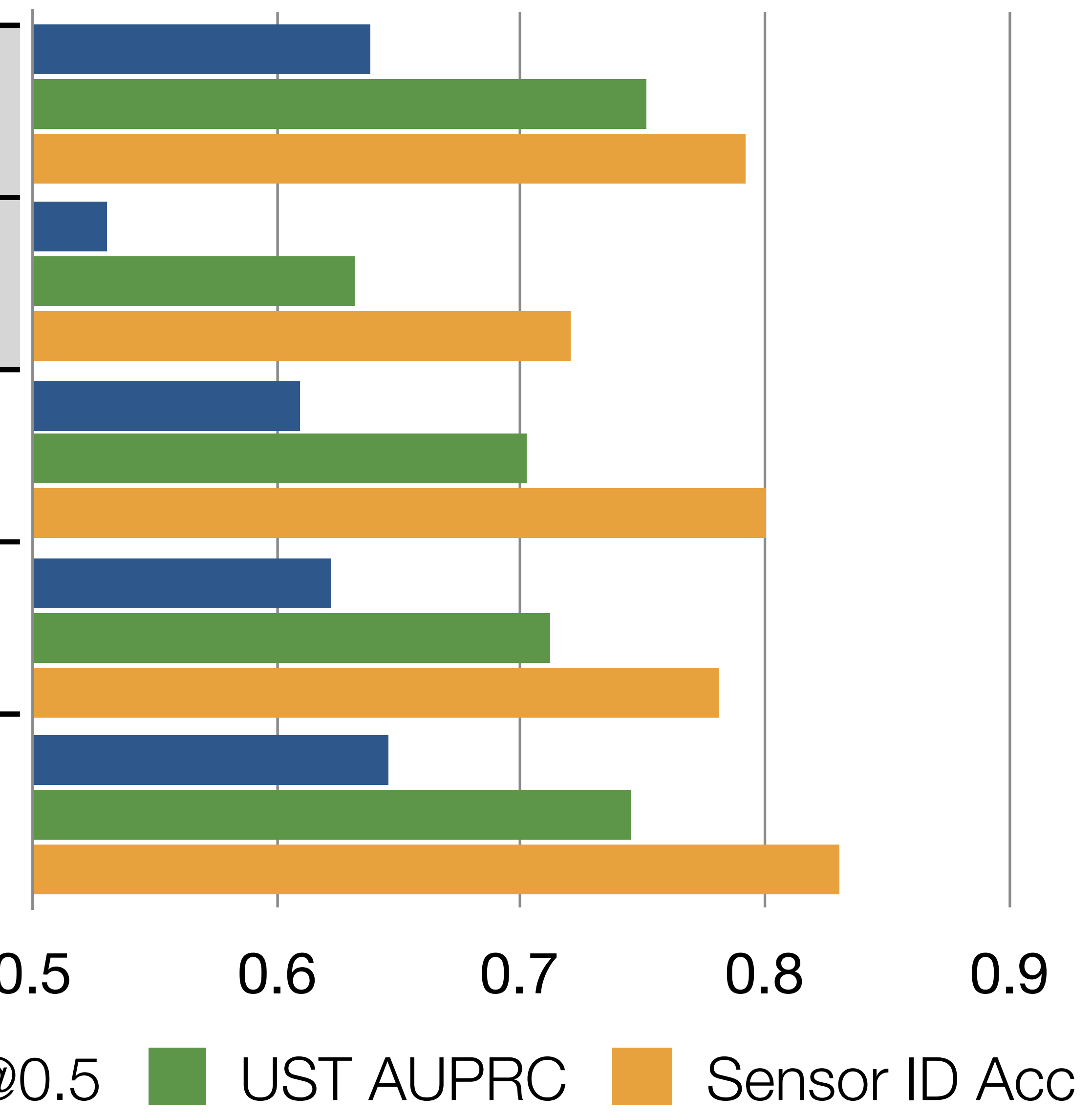
- Investigate circular regression loss formulations for von Mises distributed data
- Allow for groups of recordings with similar phase to be trained simultaneously and fused to increase the temporal signal and reduce impact of the background (hopefully reduce need for PCEN)
- Analyze the benefits of each temporal cycle and what information is encoded, and what is not
- Test TriCycle approach on other modalities

Summary

- Proposed an approach to self-supervised audio representation learning by predicting the time of recording
- First self-supervised embedding model trained on long-term temporal structure (regardless of modality)
- Able to train dataset-specific embeddings with single-modal data
- Validated approach on an urban sound tagging task, matching performance of a general state-of-the-art audio embedding
- Approach may be more general than audio, and well-suited for datasets from other sensor networks also having dense, longitudinal, timestamped data

Sensor prediction results with TriCycle

Initialization	TriCycle training	High-activity sampling	PCEN
L3	No		
Random	No		
Random	Yes		
Random	Yes	Yes	
Random	Yes	Yes	Yes



Results

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.
	(a)			(b)			(c)				(d)
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