Cross-Referencing Self-Training Network for Sound Event Detection in Audio Mixtures

Sangwook Park, *Member, IEEE* David K. Han, *Senior Member, IEEE* Mounya Elhilali, *Senior Member, IEEE*

***Abstract*—Sound event detection is an important facet of audio tagging that aims to identify sounds of interest and define both the sound category and time boundaries for each sound event in a continuous recording. With advances in deep neural networks, there has been tremendous improvement in the performance of sound event detection systems, although at the expense of costly data collection and labeling efforts. In fact, current state-of-the-art methods employ supervised training methods that leverage large amounts of data samples and corresponding labels in order to facilitate identification of sound category and time stamps of events. As an alternative, the current study proposes a semi-supervised method for generating pseudo-labels from unsupervised data using a student-teacher scheme that balances self-training and cross-training. Additionally, this paper explores post-processing which extracts sound intervals from network prediction, for further improvement in sound event detection performance. The proposed approach is evaluated on sound event detection task for the DCASE2020 challenge. The results of these methods on both "validation" and "public evaluation" sets of DESED database show significant improvement compared to the state-of-the art systems in semi-supervised learning.**

***Index Terms*—Sound event detection, semi-supervised learning, self-training, pseudo label.**

I. Introduction

AUDIO tagging summarizes an audio stream with descriptive information pertaining to the sound in terms of the location where the sound may be coming from, emotional content present, causal relationships among the sound sources, or other descriptive information. Aside from the informative nature, they can also be quite helpful in expediently retrieving or categorizing audio as the size of a typical audio database is quite large. Sound Event Detection (SED), which aims to identify sounds of interest in terms of sound category and its temporal boundaries, has been a critical technique for audio tagging. Since sounds are quite informative to understand the auditory scene such as a presence of human, animal, or any entities, and its behavior, the technique has been adopted for many applications including video analytics, baby or pets monitoring, and other surveillance systems [1]–[3]. For performing audio tagging in an environment, an SED method should be able to identify multiple sounds even when these sounds overlap with each other temporally.

With recent advances in deep learning, deep neural networks have shown outstanding improvements in SED [4]–[7]. To train a deep network for a SED task, each training sample needs to be annotated with the sound class and time boundaries of every target sound interval therein. The annotation allows the network to learn spectro-temporal characteristics of the target sound in a supervised fashion. Accurate labels and the markings of temporal boundaries are critical to train the model; however, generating them is often quite expensive and time consuming. Semi-supervised learning, which leverages extensive unlabeled data in combination with small amounts of labeled data, has been explored to resolve the issue in data collection [6], [8], [9]. In the recent challenge of the Detection and Classification of Acoustic Scenes and Events (DCASE) 2020, task 4 involves building an SED model in a semi-supervised fashion. The task provides an extensive set of unlabeled data as well as a smaller set of weakly labeled data with labels describing the sound class only [10].

Among semi-supervised techniques, self-training is an intuitive approach that is easy to understand and whose theoretical feasibility has been studied in several works [11], [12]. Self-training effectively uses the prediction of a network for a given input as a pseudo-label to further train the network. As such, the accuracy of this pseudo label has important implications on the network's performance. In an earlier work, a method of pseudo label estimation for unlabeled data and a reliability of the pseudo label were proposed [13]. The pseudo label was estimated by a probabilistic expectation of all potential labels as these probabilities were calculated based on the Bernoulli process with posterior probabilities of each class. With labeled data, reliability of the pseudo label at each training step was measured based on a binary cross entropy between true label and the estimate for the labeled data. The objective function was composed of a supervised loss for the labeled data computed from a cross-entropy between the true label and the model prediction, and an expectation loss for the unlabeled data, which was defined as a mean-squared error between the pseudo label and the prediction. The expectation loss was weighted by the reliability. Then, the model is self-trained by performing the estimation and optimization in every training step.

As an extension to the previous work, the current paper proposes a Cross-Referencing Self-Training (CRST) model. A critical issue with self-training is the self-referencing framework that has a risk of self-biasing due to the pseudo label estimated by itself. To resolve this issue, dual models composed of *Model I* trained with original data and *Model II* trained with perturbed data, are incorporated in the self-training. Each of these models is trained separately using the pseudo-label estimate of the other network. Additionally, this paper explores

S. Park is with the Department of Electronic Engineering, Gangneung-Wonju National University, Gangneung, 25457 South Korea.

D. K. Han is with the Department of Electrical and Computer Engineering, Drexel University, Philadelphia, PA, 19104 USA.

M. Elhilali is with the Department of Electrical and Computer Engineering and jointly with the Department of Psychology and Brain Sciences, Johns Hopkins University, Baltimore, MD, 21218 USA e-mail: mounya@jhu.edu.

a post-processing step to extract target sound intervals from the network prediction with a classwise thresholding and smoothing. For thresholding, the Extreme Value Theory (EVT) based threshold estimation is performed for each target class. A smoothing filter length is determined based on statistics of each target sound duration. To demonstrate effectiveness of the proposed method, experiments are performed following the protocol for the multi-target SED task in the recent DCASE challenge (DCASE2020). The result shows further improvement in the performance compared to the previous model as well as other state of the art in semi-supervised learning. The main contribution of this paper can be summarized as: 1) a novel self-training method which avoids the self-biasing issue; 2) an effective approach of selectively combining synthetic data with unlabeled or weakly labeled real world data to enable a semi-supervised training; 3) introducing a classwise post-processing involving effective estimates of threshold and duration for further improvement in SED performance.

The rest of this paper is organized as follows. Related works for semi-supervised learning are explored in the following section. The motivation of the self-training model is described in Section III, and Section IV describes the proposed methods for the cross referencing self-training model and the classwise post-processing. In Section V, experiment results performed on the DCASE challenge framework are summarized. Comparison to the challenge submission is discussed in Section VI and the conclusion is followed.

II. Related works

A. Semi-supervised learning

Semi-supervised learning aims to leverage unlabeled data to improve the performance of supervised learning with a small labeled data set. There has been a growing body of work exploring use of unlabeled data in supervised learning [14]. Usually, unlabeled data is used to learn a preliminary model about the input distribution, and the model is used for either feature extraction [15]–[17], clustering of data to assign label [18], or initialization of parameters [19]. These methods are then leveraged in a main model for classification which is trained with labeled data in supervised fashion.

Particularly, the clustering based approach assumes that two samples belonging to same cluster in observation space are likely to belong to the same class. This is known as the cluster assumption (or low-density separation assumption) and has inspired a consistency-regularization approach like PI-model [20], [21], temporal ensemble model [21], Mean Teacher (MT) model [22], and Interpolation Consistency Training (ICT) model [23]. Among these approaches, the MT model has been instrumental in pushing forth the state of the art. The MT model consists of two networks, *student* and *teacher*, and its objective function is denoted as

$$L_{MT} = BCE(y, f_\theta(x)) + \delta MSE(f_\theta(x;\eta), f_{\theta'}(x;\eta')), \quad (1)$$

where $BCE(y, f_\theta(x))$ is a classification loss implemented by a Binary Cross Entropy (BCE) between true label y and *student* prediction $f_\theta(x)$. $MSE(f_\theta(x;\eta), f_{\theta'}(x;\eta'))$ is a consistency loss implemented by a Mean Squared Error (MSE) between two predictions $f_\theta(x;\eta)$ by *student* and $f_{\theta'}(x;\eta')$ by *teacher* under a random perturbation η on each network such a rotating, shifting, or adding noise. Network parameters are represented as θ and θ' for *student* and *teacher*, respectively. This consistency loss is controlled by the δ which is usually designed with ramp-up value during the training. Both networks are constructed with the same architecture. The *student* parameters are updated by using a gradient descent method. On the other hand, parameters in the *teacher* model are updated by exponential moving average of *student* parameters over the training step. Since the averaging network tends to produce more accurate prediction than a network obtained by the gradient descent method in each training step [22], the *student* network is guided by the *teacher* network. Additionally, the consistency loss enables that the *student* network produces the same predictions even in presence of various perturbations. Once the MT model converges in training, the *student* network projects any samples belonging to the manifold constructed by the perturbations onto the same prediction.

In contrast, the ICT model differs from the MT model in that an interpolation between two inputs is considered instead of random perturbations. The objective function is denoted as

$$\begin{aligned} L_{ICT} = {} & BCE(y, f_\theta(x)) + \\ & \delta MSE(f_\theta(Mix_\lambda(x_1, x_2)), Mix_\lambda(f_{\theta'}(x_1), f_{\theta'}(x_2))), \\ & where \quad Mix_\lambda(a, b) = \lambda a + (1-\lambda) b, \end{aligned} \quad (2)$$

where λ is a random value as $0 \leq \lambda \leq 1$. The second term enables that the *student* network projects a convex set of the inputs onto another convex set of predictions for the inputs by itself. Once an ICT model converges during training, the *student* network produces similar predictions for any samples in the convex set of the inputs. This is more efficient compared to random perturbations since interpolation always generates a sample belonging to the convex set of the inputs while not all samples generated by random perturbations belong to the manifold for the actual inputs.

As an alternative approach, the present study explores a self-training method which estimates a pseudo label for unlabeled data, then updates itself for both labeled data and pseudo labeled data (unlabeled data). As a simple way to estimate pseudo label, Lee assigned a pseudo label for unlabeled data by picking up the class which shows a maximum posterior probability in the prediction by itself [11]. This method is equivalent to Entropy Regularization, which is to separate probabilistic distributions for each class by minimizing conditional entropy of the class probabilities [24], [25]. The Entropy Regularization favors a low-density separation between classes which is a principal assumption of semi-supervised learning [26]. With this background, the self-training with pseudo label for the best class was applied to object detection and hyperspectral image classification [27], [28]. Instead of picking up the best class, a pseudo label was estimated based on probabilistic expectation of potential labels due to the multiple target scenario for SED task [13]. Recently, Wei, et al studied a theoretical background of the empirical successes in self-training with deep learning [12].

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3178591, IEEE Transactions on Multimedia

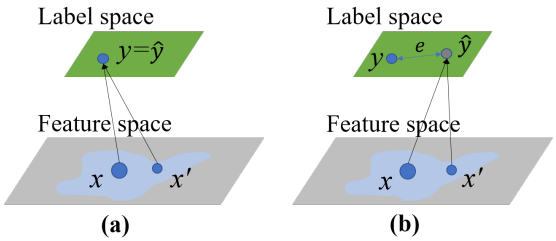


Fig. 1. An SED model projects inputs in feature space onto label space where x and x' are input features. y is a true label while $\hat{y}$ is the prediction (a) Strategy of consistency-regularization method, (b) Limitation of consistency-regularization.

B. Semi-supervised learning in sound event detection

The DCASE challenge provides a framework of semi-supervised learning for sound event detection: Both the baseline implementation and the training database including unlabeled data. In the recent DCASE2020 challenge, the baseline was implemented with a Convolutional Recurrent Neural Network (CRNN) and MT model for network architecture and semi-supervised learning respectively [10]. The training database was composed of three subsets: a strong labeled dataset S, a weakly labeled dataset W, and an unlabeled dataset U. The strong label describes target sound class as well as time boundaries for each target sound interval. For the weakly labeled data, its label describes a list of target sound classes without time boundaries. Unlabeled data has neither the sound class nor the time boundary. With these three types of training data, the objective function (1) is modified as follows:

$$L_{MT} = \sum_{x \in S} BCE(f_\theta(x), y^s) + \sum_{x \in W} BCE(E[f_\theta(x)], y^w) + \delta \sum_{x \in S,W,U} MSE(f_\theta(x), f_{\theta'}(x')). \tag{3}$$

where y^s and y^w are label for strong labeled data and weakly labeled data, respectively. $E[.]$ is expectation over the time. Note that x' is generated by adding Gaussian noise to x with 30 dB Signal to Noise Ratio (SNR) condition.

Miyazaki et al. won in the DCASE2020 challenge with an integration of strategies including a new architecture, data augmentation, classwise post-processing, and fusion [29] while they employed the MT approach for semi-supervised learning. Koh, et al. suggested a Shift Consistency Training (SCT), which makes the network to produce consistent predictions for time-shifted inputs, and feature pyramid network to predict temporal label for weakly data [30]. Additionally, they explored ICT approach for SED task. With an integration system of MT, ICT, SCT, and feature pyramid, they have achieved best performance on their own. Kim, et al. proposed a modified CRNN network, which is using more filters and skip-connections with attention module, as well as modified loss function, which is a cross entropy between network prediction and pseudo label estimated by an weighted sum of predictions by the modified network and output of the challenge baseline [31], [32]. With data augmentation based on time-frequency masking and interpolation, their system outperformed the challenge baseline.

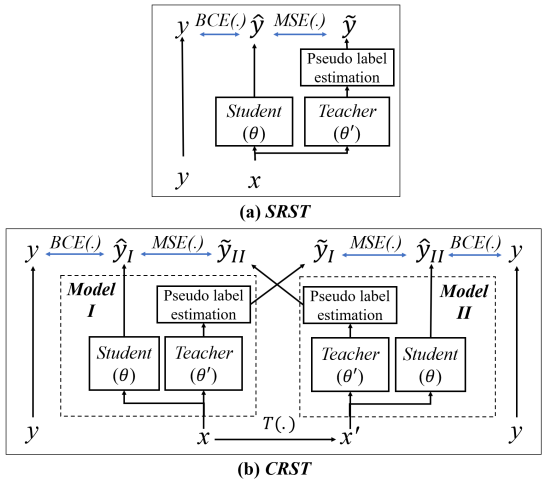


Fig. 2. Diagram for self-training framework where y, $\hat{y}$, and $\tilde{y}$ is a true label, *student* network's prediction, and pseudo label, respectively. x' represents manipulated data from the original x by a transformation function $T(.)$. (a) Self-Referencing Self-Training (SRST) model, (b) Cross-Referencing Self-Training (CRST) model.

III. Motivation of Proposed Semi-Supervised Training Method

Fig. 1(a) shows a concept of consistency-regularization approach. The approach allows the model producing a consistent prediction for all neighbors of one input in training set. Based on the cluster assumption, it is able to improve the performance in the classification task. In addition, it allows unlabeled data to be incorporated in supervised learning for the consistent prediction as calculated by the consistency loss. From the perspective of semi-supervised learning however, the consistency loss can be thought of as an estimation of the difference between two predictions; which in of itself could result in convergence on the wrong label as shown in Fig. 1(b). These errors may have a great effect on the model performance since unlabeled data is generally far bigger in size than labeled data [33].

In a previous work, a probabilistic expectation of potential labels was proposed as a pseudo label for unlabeled data [13]. In order to mitigate the issue of erroneous label mapping shown in Fig. 1(b), we defined ϵ as the mean squared error between the pseudo label and network prediction, and the expectation error is then minimized. This method was shown to outperform the MT model on a SED task. However, this approach is not limitation-free, and has a risk of self-biasing because the pseudo label is estimated by itself.

As an extension of the previous work, this paper proposes a Cross-Referencing Self-Training (CRST) model to mitigate the self-biasing issue. Fig. 2 contrasts the two approaches. Fig. 2a depicts the previous model, named Self-Referencing Self-Training (SRST). Fig. 2b depicts the proposed approach, the CRST model, which consists of two self-training models, where each model is trained on either original or manipulated data with the pseudo label estimated by the other model. The cross-referencing of a pseudo label estimated by the other model is a key point of this dual structure. It is able to avoid the self-biasing issue based on the assumption that those networks are independently trained on a different version of data.

IV. Proposed method

The implementation of proposed model can be found in *http://github.com/JHU-LCAP/CRSTmodel*.

A. Pre-processing

An audio clip is processed to a 16kHz mono-channel audio waveform by resampling and averaging left and right channels for multi-channel audios. The audio waveform is converted to a spectrogram by performing Short Time Fourier Transform (STFT) with 2048-points frame length and 255-points hop size. Then, a log-Mel spectrogram is obtained by performing frequency integration with 128 Mel-filters spanned 0 to 8kHz frequency domain and logarithm function.

$$\begin{aligned} P[n,m] &= \sum_k S[k,m] \times f^n_{mel}[k], \\ x[n,m] &= log(max(P[n,m]^2,\ \epsilon^2)), \end{aligned} \tag{4}$$

where x is a log-Mel spectrogram while S is a spectrogram based on STFT. f^n_{mel} is Mel filter for the n^{th} channel. n, m, and k are indices to represent Mel filter channel, frame, and frequency bin, respectively. Note that the ϵ is set to $1.0E-5$ in order to prevent negative infinite value by the logarithm function. The different version of data, x' in Fig. 2 is generated by adding the Gaussian noise to the log-Mel spectrogram with a given Signal to Noise Ratio (SNR) condition. Note that audio length was considered up to 10 second so that zero-padding or cutting is performed for shorter or longer audios than 10 second. The SNR condition to generate manipulated data was set to 30 dB from the challenge baseline [10].

B. Network architecture

From the challenge baseline for SED task, CRNN is applied to both networks in *Model I* and *Model II*. The CRNN is composed of two stages: Convolutional Neural Network (CNN) to compress the log-Mel spectrogram into acoustic features and Bidirectional Gating Recurrent Units (BGRUs) to capture temporal relations among the compressed features by the CNN. The first stage is composed of seven convolution layers and seven averaging pooling layers. Each convolution layer uses a Gated Linear Unit (GLU) for a nonlinear activation. The GLU controls the selection of critical features in order to capture informative characteristics among the target sounds by using a self-gating function described by

$$c' = GLU(c) = Linear(c) \times \sigma(c), \tag{5}$$

where c and c' represents a result of convolution and GLU, respectively. $Linear(.)$ and $\sigma(.)$ is a linear transformation and a sigmoid function for the gating, respectively. In training phase, batch normalization and dropout techniques are applied to before and after performing the GLU, respectively.

The following stage consists of a Double-layered Bidirectional Gated Recurrent Unit (BGRU). The output of the CNN represents compressed features along to output channels across the time. These features are fed into the BGRU to learn the temporal characteristics of target sounds in onset and offset edges. During the training phase, dropout is applied to the end of each unit. Additionally, a linear layer with a sigmoid activation is added on the top of the BGRU to represent a presence probability of target sounds. As a result, a set of likelihood probability across the target sounds over the time is outputted by the second stage. More detailed setup for this architecture can be found in Appendix A or implementation of the DCASE2020 challenge baseline for task 4 [10].

C. Objective function

With the idea of cross-referencing, objective function is designed to resolve self-bias issue. To train the *student* network in each model with the three types of data, strong labeled, weakly labeled, and unlabeled, in supervised fashion, the objective function consists of a classification error and an expectation error with a reliability of pseudo label. The classification error is defined by a BCE between the network prediction and strong label y^s or weak label y^w while a MSE between the prediction and pseudo label $\tilde{y}$ is used for the expectation error (6).

$$\begin{aligned} L_I =& \sum_{x \in S} BCE(f_I(x), y^s) + \sum_{x \in W} BCE(E[f_I(x)], y^w) \\ &+\gamma^s_{II} \sum_{x \in U} (MSE(f_I(x), \tilde{y}_{II}) + \gamma^w_{II} \sum_{x \in W} (MSE(f_I(x), \tilde{y}_{II}), \\ L_{II} =& \sum_{x' \in S} BCE(f_{II}(x'), y^s) + \sum_{x' \in W} BCE(E[f_{II}(x')], y^w) \\ &+\gamma^s_{I} \sum_{x' \in U} (MSE(f_{II}(x'), \tilde{y}_{I}) + \gamma^w_{I} \sum_{x' \in W} (MSE(f_{II}(x'), \tilde{y}_{I}), \end{aligned} \tag{6}$$

where L_i is the objective function for training *Model i* whose prediction is denoted as $f_i(.)$. x' is generated by adding Gaussian noise to original data x with 30 dB Signal to Noise Ratio (SNR) condition. $E[.]$ is an averaging operator over the frames. Note that the weak label y^w is a vector indicating target sound class which happened in the audio clip while the strong label y^s is a matrix stacking the vectors for every frame. γ^s_i and γ^w_i are the reliability of pseudo label estimated in *Model i* with strong labeled data and weakly labeled data, respectively. During the training phase, the parameters for both models, *Model I* and *Model II*, are separately optimized with each objective function (6).

1) Pseudo label estimation: Considering a scenario for multiple sound detection, a label indicating the presence of target sounds in a frame is represented to a zero vector for non-target, one-hot vector for single target, or a many-hot vector for multiple targets at least two. A pseudo label for a frame is estimated to a probabilistic expectation of those potential labels. By performing this estimation for every frame, the pseudo label of unlabeled audio clip is represented to a matrix like the strong label. Since the pseudo label has a expectation value not a binary value, the MSE criterion is more appropriate for the expectation loss in (6) than the BCE. With this concept, the pseudo label can be estimated as

$$\tilde{y} = \Sigma^K_k\ \Sigma^{N_k}_n\ p^k_n\ l^k_n, \tag{7}$$

where k is the number of concurrent events in each frame and n is an index for the case of choosing k-sounds of total target

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3178591, IEEE Transactions on Multimedia

sounds. l_n^k is a label vector expressed by a summation of delta functions like $l_{n:\{i,j\}}^2 = \delta_i + \delta_j$ for events i and j ($(k = 2)$. p_n^k is a probability of the label l_n^k, K is maximum number of concurrent events, and $N_k = C!/(k! \times (C-k)!)$ is the number of potential labels under the k and total number of target sound classes C. Note that this estimation is performed for every frame even though the frame index is omitted for brevity.

Based on the fact that the *teacher* network produces more accurate predictions [22], the probability of a label vector is calculated with the *teacher* prediction in each model. Based on the Bernoulli process for activation of each target sound, the probabilities p_n^k are calculated depending on the number of concurrent events k as

$$\begin{aligned} k=0, \quad & p_{n:\{\}}^0 = \frac{1}{N}\Pi_q\ (1-\hat{y}'_q), \\ k=1, \quad & p_{n:\{i\}}^1 = \frac{1}{N}\hat{y}'_i\Pi_{q\neq i}\ (1-\hat{y}'_q), \\ k=2, \quad & p_{n:\{i,j\}}^2 = \frac{1}{N}\hat{y}'_i\hat{y}'_j\Pi_{q\neq i,j}\ (1-\hat{y}'_q), \\ k=3, \quad & p_{n:\{i,j,h\}}^3 = \frac{1}{N}\hat{y}'_i\hat{y}'_j\hat{y}'_h\Pi_{q\neq i,j,h}\ (1-\hat{y}'_q), \\ & \ldots, \end{aligned} \tag{8}$$

where N is a normalization factor as $N = \Sigma_k^K\ \Sigma_n^{N_k}\ p_n^k$. Note that the prediction by *teacher* network of *Model I* (*Model II*) is applied to the pseudo label estimation for training *Model II* (*Model I*).

The estimation of a pseudo label in every frame for all unlabeled data introduces a heavy computational load in the calculation for all potential labels. To reduce this computation, the number of concurrent events k is considered up to 2 based on previous work [13]. The probabilities for multi-sound labels are calculated using a dynamic programming technique (9).

$$\begin{aligned} k=0, \quad & P^0 = log(p_{n:\{\}}^0), \\ k=1, \quad & P_i^1 = P^0 + log(\hat{y}'_i) - log(1-\hat{y}'_i), \\ k=2, \quad & P_{\{i,j\}}^2 = P_i^1 + P_j^1 - P^0, \end{aligned} \tag{9}$$

2) Reliability of pseudo label: The prediction by the *teacher* network is obviously unreliable at the beginning of training. Even at later stages of training, the pseudo label is still an estimate based on the prediction. Therefore, the expectation error is weighted by the reliability of pseudo label to adjust the contribution of the error on training. In this study, the Jensen Shannon Divergence (JSD), which is bounded in [0,1], is considered to calculate the reliability γ^s and γ^w of the pseudo label with strong labeled and weakly labeled data, respectively (10).

$$\begin{aligned} \gamma^s = & \ \omega \times \frac{1}{N^s}\sum_{x\in S}(1 - JSD(\tilde{y}||y^s)), \\ \gamma^w = & \ \omega \times \frac{1}{N^w}\sum_{x\in W}(1 - JSD(E[\tilde{y}]||y^w)), \\ & where\ \omega = 3.0exp(-5(1-t/T)^2), \\ & JSD(a||b) = KLD(a||m)/2 + KLD(b||m)/2, \\ & m = (a+b)/2, \end{aligned} \tag{10}$$

where ω is a ramp-up parameter with an index of training step t and maximum number of the steps T, N^s and N^w is the number of strong labeled data and weakly labeled data, respectively. KLD is a Kullback Leibler Divergence. At the beginning of training, the expectation error (6) remains small due to the ω. In later stage of training, the reliability relies on the similarity between pseudo label for labeled data and true label.

D. Post-processing

Class imbalance in training dataset is another issue often encountered in semi-supervised learning. The sparsity of the minority classes in the training set minimizes their contribution to the objective function resulting in a bias toward the majority class [34]. Dynamic sampling [35] or data augmentation [36] for minority class data is an effective way to resolve this issue. In the scenario of semi-supervised learning, however, it is hard to use either one because those methods need class label for all training dataset. Instead, this paper explores classwise post-processing to extract target sound intervals from class posterior probabilities over time (i.e. *student* output). Once the *student* model converges after training, the network's output exhibits different distribution of posteriors to each target class (Fig. 3). Thus, post-processing composed of thresholding and smoothing is performed with optimized classwise parameters, threshold and smoothing length.

1) Threshold estimation: In each target class, a threshold is estimated based on the Extreme Value Theory (EVT) [37]. Once a network's training converges, samples used in threshold estimation are collected by applying logit function, $logit(x) = log(\frac{x}{1-x})$, to the network prediction responding to weakly labeled data. In a threshold estimation for "*Speech*" class, for example, audio clips which have the "*Speech*" sounds in weakly labeled dataset are used to collect network predictions for the "*Speech*" class. Since a target sound duration is typically shorter than 10 second, the samples can be categorized into two clusters for target and non-target (Fig. 3). These clusters are separated based on the Expectation and Maximization (EM) clustering [38]. Threshold estimation is performed with samples belonging to the target cluster, which has a greater mean value than the other since the network has been trained. To apply the EVT to the samples, the target samples are reversed by multiplying -1. Motivated in [39], Cumulative Distribution Function (CDF) of the reversed samples, $F(x)$, is defined as

$$\begin{aligned} F(x) = (1 - Pr(x \leq u))F_u(x-u) + Pr(x \leq u), \\ Pr(x \leq u) = \frac{N-n}{N}, \end{aligned} \tag{11}$$

where u is a predefined threshold to extract extreme samples which are greater than the predefined threshold, $Pr(x \leq u)$ is a probability of a set of samples x, which are less than u. N is the total number of samples and n is the number of extreme samples. In this study, u was defined to the value that satisfies $Pr(x \leq u) = 0.9$. Tail distribution, i.e. CDF of the extreme samples $F_u(x-u)$, is modelled with a Generalized Pareto Distribution (GPD) [40] as

$$G(z) = 1 - (1 + c\frac{z}{a})^{-1/c}, \tag{12}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3178591, IEEE Transactions on Multimedia

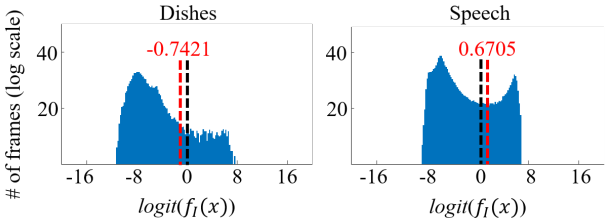


Fig. 3. Histograms for posterior distribution in two targets, *Dishes* and *Speech*. Once a model is converged in training, posteriors were calculated on weakly labeled data. Red dotted lines show optimal thresholds ($-t_\alpha$) for each class while black dotted line (at 0.0 in logit domain) represents a global threshold for all targets.

where $G(z)$ is a CDF of the extreme samples with $z = x - u$. a and c are tuning parameters optimized to maximize log-likelihood for all extreme samples

$$\sum_{i=1}^{n} log(g(z_i)) = -nlog(a) - \frac{1+c}{c}\sum_{i=1}^{n} log(1 + c\frac{z_i}{a}), \quad (13)$$

where $g(z)$ is a probability density function of $G(z)$. Note that Nelder-Mead Simplex method [41] is applied to find optimal a and c. With the parameters in (11) and (12), a threshold with a given parameter α is estimated as

$$t_\alpha = u + \frac{a}{c}((\frac{N\alpha}{n})^{-c} - 1), \quad (14)$$

where α means a theoretical probability of false negative. Because the t_α in (14) is derived for reversed samples in logit domain, the threshold applied to post-processing is finally obtained by $\sigma(-t_\alpha)$.

For instance, Fig. 3 shows histograms of posterior probabilities calculated on weakly labeled data and optimal thresholds in two classes. *Dishes* sounds have an issue of imbalance between target and non-target frames because events are too short compared to whole duration. Thus, the distributions are biased towards negative values which increases likelihood to belong to a non-target frame. In this case, the threshold (marked with a black line) needs to be moved to left-side in order to enhance detection rate. On the other hand, *Speech* seems to be free from this issue as shown in its distribution which is nearly zero-centered. The threshold estimation method produces a small positive shift of the global threshold, which enables to reduce false positive.

2) Smoothing with median filter: Since thresholding based detection is performed on each of frames, a low-pass filtering is needed to determine sound intervals which are slowly changed in nature. So, a smoothing is performed by applying median filter to the frames detected in the previous step. It is needed to optimize the filter length for a precise time interval since the filter length directly affects on the times at rising and falling edges. In this work, the filter length is determined by β % of average of sound duration. The average of sound duration is estimated with the number of frames resulted in thresholding with respect to weakly labeled data.

V. Experiment

A. Database

In order to evaluate the proposed system, the DESED database was used [42]. This database contained 10-target sounds for SED task: *Alarm_bell_ringing*, *Blender*, *Cat*, *Dishes*, *Dog*, *Electric_shaver_toothbrush*, *Frying*, *Running_water*, *Speech*, and *Vacuum_cleaner*. The database included a training set composed of *Synthetic:training*, *Real:weakly labeled*, and *Real:unlabeled* as shown in TABLE I. To synthesize strong labeled data, background sounds were extracted from SINS [43], MUSAN [44], or Youtube; and target sounds were obtained from freesound [45]. In this training set, two points are worth highlighting: 1) Synthetic data is considered as strong labeled data instead of marking ground truth on real recordings. 2) The number of unlabeled data is much larger than the number of labeled data. For the post-processing technique proposed in this study, statistics were collected with weakly labeled data. And, two subsets, *Real: validation* and *Real: public evaluation* were used for performance assessment.

TABLE I
Database

DESED set	label type	# of audio clips
Synthetic:training	strong labeled	2,595
Real:weakly labeled	weakly labeled	1,578
Real:unlabeled	unlabeled	14,412
Real:validation	strong labeled	1,168
Real:public evaluation	strong labeled	692

B. Evaluation criterion

The assessment was performed with a f-score, which is a harmonic mean of precision and recall. The precision was calculated as the ratio of true-positive intervals to total detected intervals. The recall was equal to a detection rate, the ratio of true-positive intervals to target intervals. Those measures were calculated for each class. Based on the evaluation protocol for a SED task in the DCASE2020 challenge [46], in this work, a detected interval was considered as true-positive if the interval satisfies three conditions: 1) onset time of the interval precedes earlier than the truth as less than a 200 ms. 2) offset time of the interval delays the truth as less than 200 ms. And 3) a sound class of the interval should be matched to the true class. Note that the 200 ms margin in both time boundaries was considered to prevent slicing the sound in the middle.

C. Models for SED task

To demonstrate the effectiveness of the proposed method, CRNN network (See, Section IV-B) was trained in different ways using a 5 times cross-validation method.

1) Supervised learning: This approach was considered separately for strong-labeled data only, or both strong and weakly labeled data. In the first model, only strong labeled data (synthetic data) was used for training. The loss function was defined as $L = \Sigma_{x \in S} BCE(f_\theta(x), y^s)$. In the second model, both strong and weakly labeled data were used for training. The loss function was defined as $L = \Sigma_{x \in S} BCE(f_\theta(x), y^s) + \Sigma_{x \in W} BCE(E[f_\theta(x)], y^w)$.

2) Consistency regularization: As a state of the art in semi-supervised learning, MT and ICT models were considered in this category. For training a MT model, the DCASE2020 challenge baseline whose objective function is denoted in (3)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3178591, IEEE Transactions on Multimedia

TABLE II
PERFORMANCE ASSESSMENT WITH CLASS AVERAGING F-SCORE

		Validation		Public evaluation	
		Global threshold	Classwise threshold	Global threshold	Classwise threshold
Supervised learning	Strong labeled only	17.65 ± 0.70		25.15 ± 1.07	
	Strong & Weakly labeled	28.67 ± 2.82		31.32 ± 3.51	
Consistency Regularization	MT (DCASE baseline)	34.04 ± 1.47	36.37 ± 1.18	38.77 ± 1.59	41.09 ± 2.37
	ICT (our implementation)	35.34 ± 1.94	38.60 ± 1.99	37.19 ± 3.06	39.17 ± 1.90
Self-training	SRST [13]	36.36 ± 0.76	37.65 ± 0.91	37.96 ± 1.58	39.55 ± 1.58
	SRST + aug.	35.28 ± 1.93		36.21 ± 1.08	
	CRST	**37.89 ± 1.01**	**39.76 ± 1.90**	**40.19 ± 1.59**	**42.81 ± 2.03**

was used. With this implementation, ICT model was trained with modified objective function as

$$L_{ICT} = \sum_{x \in S} BCE(f_\theta(x), y^s) + \sum_{x \in W} BCE(E[f_\theta(x)], y^w) + \delta MSE(f_\theta(Mix_\lambda(x_1, x_2)), Mix_\lambda(f_{\theta'}(x_1), f_{\theta'}(x_2))). \quad (15)$$

3) Self-referencing self-training: The previous version of self-training, SRST model, was evaluated as well. The loss function was set to

$$L_{SRST} = \sum_{x \in S} BCE(f_\theta(x), y^s) + \sum_{x \in W} BCE(E[f_\theta(x)], y^w) + \gamma \sum_{x \in W,U} (MSE(f_\theta(x), \tilde{y}), \quad where \;\; \gamma = min(\omega / BCE(y^s, \tilde{y}), 5.0), \quad (16)$$

where $\tilde{y}$ is pseudo label estimated by itself and ω is defined in (10). And another model (SRST+aug.) was trained with data augmentation based on adding Gaussian noise with 30 dB SNR condition. Finally, the cross-referencing self-training approach, CRST model, was evaluated.

D. Performance in class-averaging f-score

TABLE II shows class averaging f-scores for each model which are the mean and standard deviation over the 5 times repetition. To investigate the effect of post-processing, f-scores obtained using the global threshold (0.5) and median filter (445ms) for all targets are summarized in the column of "Global". Note that the global parameters were determined based on the challenge baseline. Results in "Classwise" on "validation" were obtained by performing the proposed post-processing with the best parameters that were heuristically determined in searching within the intervals from 0.0002 to 0.1 with 10 steps in log-scale for α in (14) and from 5% to 100% with 20 steps in linear scale for β in the estimation of filter length. Then, these best parameters were applied to the test on "public evaluation" set for each model.

Since the strong labeled dataset is composed of synthetic data produced by mixing target sounds and a background sound, it could contain artifacts such as unnatural transition in target sound boundaries and unnatural causality among target sounds. In training the synthetic data, a model could rely on these artifacts to detect target sounds. The results for two supervised models in Table II suggest that this issue could be alleviated by including real data (weakly labeled data) in training. The table also demonstrates that using unlabeled data in network training is effective to enhance the SED performance. In the evaluation on "validation" set, self-training methods except "SRST+aug." outperform the methods of consistency regularization if global post-processing is used. With 5% significant in Welch's t-test [47], particularly, the CRST model improves the f-score significantly compared to MT model (*p-value*=0.0013), ICT model (*p-value*=0.0313), and the SRST model (*p-value*=0.0265). If the classwise post-processing is performed, all of the f-scores are improved about 2.0-3.0% compared to the results in the column of "Global". The Welch's t-test on these results confirms that the CRST model shows a significant improvement compared to MT model (*p-value*=0.0096) while the CRST model averagely outperforms other two models as shown in the table (in t-test with SRST model: *p-value*=0.0566, with ICT model: *p-value*=0.3788).

In the second evaluation on the "public evaluation" set, the CRST model outperforms other models as well. In this evaluation, the MT model shows the second best in the average of f-scores over the repetition. In T-test with the results of global post-processing, the p-values are 0.1931, 0.0869, and 0.0560 in comparison to the MT model, the ICT model, and the SRST model, respectively. If the classwise post-processing is performed, the CRST model shows a significant improvement compared to the ICT model (*p-value*=0.0192) and the SRST model (*p-value*=0.0221). On the other hand, the CRST model averagely outperforms than the MT model (*p-value*=0.2536).

From these evaluations, the CRST model results in more accurate detection of sound intervals with stable performance on both datasets. Additionally, the proposed post-processing enables to improve the performance about 2.0-3.0% in class-averaging f-score.

E. Investigation of classwise performance

To explore f-scores in individual class, classwise f-scores for semi-supervised models are represented in Fig. 4. In the classwise comparison on "validation" set (Fig. 4(a), results of classwise post-processing), the CRST model has reached the best performance for five classes and the second best for two classes in the mean of 5 times repetition. With 5% significant in the T-test, particularly, the model shows a significant improvement in *Electric_shaver_toothbrush* and *Speech* compared to the MT model. Compared to the ICT model, significant improvement could be found in *Dog* and *Speech*. And *Dishes*, *Dog*, and *Speech* are significantly improved compared

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3178591, IEEE Transactions on Multimedia

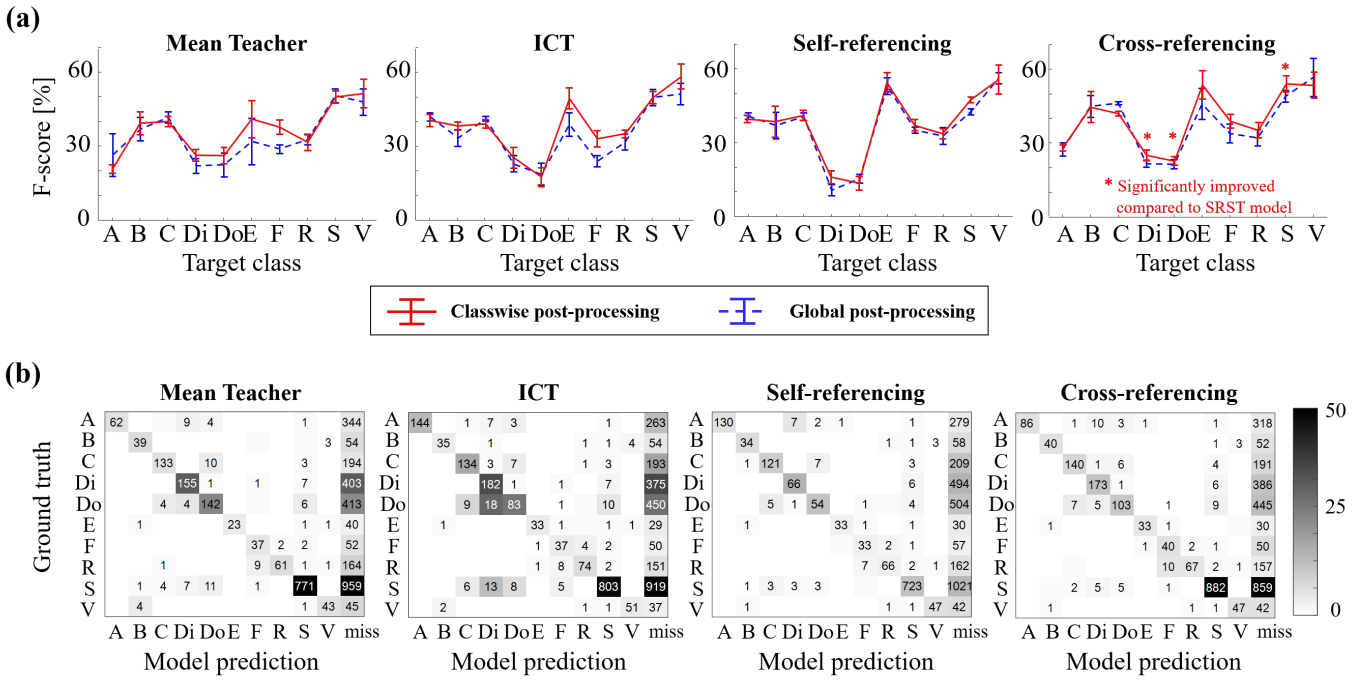


Fig. 4. Classwise performance on "validation" set. (a) Classwise fscores of two post-processing methods in four semi-supervised models. The results of classwise post-processing are marked as red solid line while blue dotted line is for the results of global post-processing. The error bar means the standard deviation over the 5 times repetition. (b) The most left table shows the number of sound intervals on "validation" set for each class. Then, four matrices show a confusion in classification for detected intervals which are matched to the truth in time boundaries. The numbers are the mean over the 5 times repetition and the standard deviation is represented to background light in black and white.

to the SRST model. In a classwise performance, the SRST model shows the biggest variation across the target classes among the four models. The f-scores of the SRST model are averagely better than other models in *Alarm_bell_ringing* and *Electric_shaver_toothbrush*. Among the four models, the worst f-score for *Dishes* and *Dog* could be found in the SRST model as well. It is difficult to investigate what happen exactly in these classes during the training due to the lack of labels in the training data. One of potential reasons is that the SRST model was biased by a pseudo label that was estimated incorrectly due to some reasons such as a noise or an overlapping effect. Once the model miss *Dishes* sounds, the model never detect the *Dishes* sounds because the pseudo label is unable to give any information for the *Dishes*. Therefore, the *Dishes* class would be getting worse and worse, on the other hand, the *Electric_shaver_toothbrush* class would be getting better and better because it is able to reflect more samples for the *Electric_shaver_toothbrush* class in training. This issue has been resolved with the cross-referencing framework as shown in the result of the CRST model. To investigate the detection results of each model, each confusion matrix for the results based on classwise post-processing is represented on Fig. 4(b). With detected intervals that have matched to the truth in time boundaries, the confusion matrix was built by counting the number of the intervals in classification. To give the information about missing intervals in time boundaries, the most right column in each confusion matrix shows the number of the missing intervals by the model. Note that the numbers on each confusion matrix are the mean value over the repetition and the standard deviation is represented to background light. In both the ICT and SRST model, the f-score in *Dog* is the worst among the targets. In case of the ICT model, the worst result is owing to a confusion among the classes. On the other hand, inaccurate detection in time is the reason of the SRST model as in that only 65 intervals were detected with correct time boundaries.

The results performed on "public evaluation" set are summarized in Fig. 5. As shown in Fig. 5(a), the CRST model reached the best performance in six classes and the second best in two classes. In this assessment, the MT model outperforms the ICT and SRST model as shown in Table II. In a classwise comparison, the ICT and SRST model outperforms the MT and CRST models in *Alarm_bell_ringing* while these models still have in trouble to detect *Dishes* and *Dog* compared to other two models. And the CRST model shows a significant improvement in *Speech* compared to all other models. Similarly, the confusion matrices in this assessment are represented in Fig. 5(b). The trend is mostly consistent with the results on "validation" set except in *Cat*, *Dog* and *Electric_shaver_toothbrush*. The sounds of "Cat" and "Dog" have a big variation depending on species, size, or age, which could explain the different performance between the "validation" and "public evaluation" sets. It can also be noted that the number of *Cat* and *Dog* intervals is less than the number of the sounds in "validation set". On the other hand, more sounds for *Electric_shaver_toothbrush* are included in the "public evaluation" set, and the performance in this category shows worse than the f-score in "public evaluation" set. Note that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3178591, IEEE Transactions on Multimedia

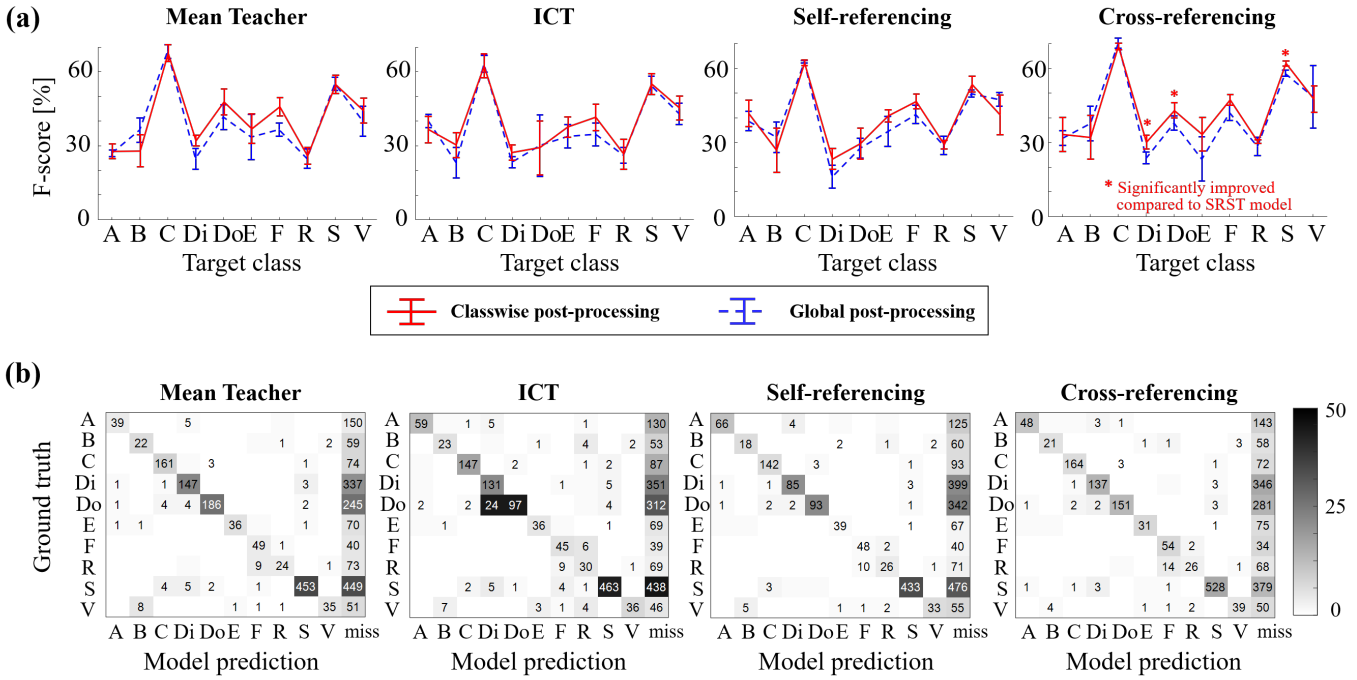


Fig. 5. Classwise performance on "public evaluation" set. (a) Classwise fscores of two post-processing methods in four semi-supervised models. The results of classwise post-processing are marked as red solid line while blue dotted line is for the results of global post-processing. The error bar means the standard deviation over the 5 times repetition. (b) The most left table shows the number of sound intervals on "validation" set for each class. Then, four matrices show a confusion in classification for detected intervals which are matched to the truth in time boundaries. The numbers are the mean over the 5 times repetition and the standard deviation is represented to background light in black and white.

the total number of target intervals in this evaluation can be found by a summation of each row of the confusion matrix.

TABLE III
PERCENTAGE OF THE NUMBER OF CONCURRENT SOUND INTERVALS IN BOTH TEST DATASETS

	Validation			Public evaluation		
	$k=1$	$k=2$	$k>2$	$k=1$	$k=2$	$k>2$
A	82.14	17.38	0.48	67.35	30.10	2.55
B	61.46	38.54	0	72.62	26.19	1.19
C	89.44	10.56	0	87.08	11.67	1.25
Di	41.09	48.15	10.76	37.91	56.56	5.53
Do	81.23	18.6	0.17	77.33	21.54	1.13
E	46.15	53.85	0	44.44	54.63	0.93
F	11.7	61.7	26.6	24.44	58.89	16.67
R	65.4	31.65	2.95	67.89	30.28	1.83
S	55.25	42.13	2.62	41.62	54.66	3.72
V	69.57	30.43	0	82.29	17.71	0
Total	62.18	34.47	3.35	55.37	41.27	3.36

F. Exploration of maximum number of concurrent events

In order to reduce a computational load in pseudo label estimation, in this study, the number of concurrent events, k, was considered up to 2 ($K = 2$) so that total 56 potential labels ($= 1 + 10 + 45$ for $k = 0$, $k = 1$, and $k = 2$, respectively) were used to estimate pseudo label. According to the previous study in SRST model [13], the class averaging f-score was saturated at $K = 2$ because the case, which three or more target sounds happen at a time, is unusual in practical environments. With both test datasets, sound intervals that are overlaid with other target sound were counted and the results are summarized in TABLE III. In total, the ratio of cases of three or more concurrent sounds, is less than 5 % in both datasets. Thus, the preset parameter for the maximum number of concurrent sounds is acceptable assumption in the pseudo label estimation.

G. Effect of preset parameters in post-processing

As shown in the results (Fig. 4(a) and 5(a)), the performance is mostly improved by applying the classwise post-processing compared to the results of the global post-processing. For performing the classwise post-processing, preset parameters, a theoretical false negative rate α and a ratio to the average of sound duration β, are heuristically decided in searching within the intervals from 0.02% to 10% with 10 steps in log-scale for α in (14) and from 5% to 100% with 20 steps in linear scale for β in the estimation of filter length. For the CRST and ICT (the second best model on "validation" set as in TABLE II), class averaging f-scores depending on the parameters are represented in Fig. 6. Fig. 6(a) shows class averaging f-score depending on theoretical false negative rate α in (14). In both panels, the red line represents the result based on the global post-processing. The gray region and the error-bar represent a 95 % confidence interval for the mean. Note that the β was set to 45% and 25% for ICT and CRST model, respectively. According to the results, the f-score has been improved significantly on 0.32 and 0.64 % for both models. If the α is too big or small, it is unable to improve the performance due to the false results such as FPs or FNs. The f-scores depending on β are represented in Fig. 6(b). Note that the α was set to 0.32% and 0.64% for ICT and CRST model.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3178591, IEEE Transactions on Multimedia

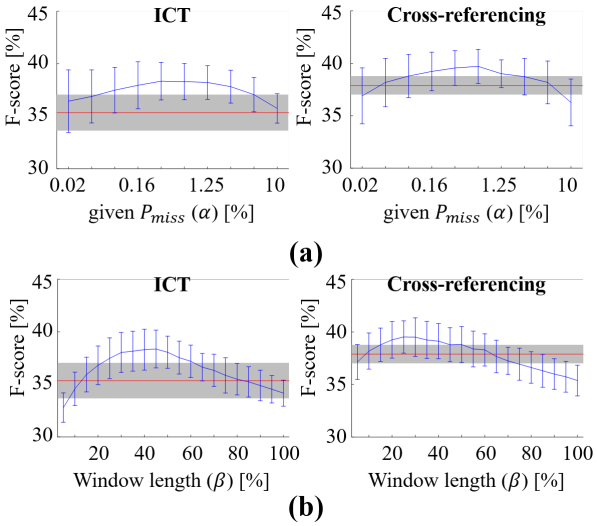


Fig. 6. Class averaging f-score based on classwise post-processing with different preset parameters. The red line on each panel represents the fscores based on global post-processing and the gray region and error bar represent a 95 % confidence intervals for the mean value. (a) depending on parameter α in ICT and CRST model with $\beta = 45\%$ and $\beta = 25\%$. (b) depending on parameter β in ICT and CRST model with $\alpha = 0.32\%$ and $\alpha = 0.64\%$.

As shown in results, the length of smoothing filter is a critical parameter effecting on the performance since the smoothing makes an early or a delay on time boundaries of detected intervals. If a short length filter is used in the smoothing, it is unable to remove very short intervals resulted in the thresholding. Thus, the smoothing remains the FPs, short-time intervals due to a noise, and makes small precision in the evaluation. On the other hand, it could merge two intervals which are close to each other in time when a long length filter is used. In this case, the smoothing could reduce TPs and make FPs more because the merging interval would be decided to FPs due to the mismatching in time boundaries.

H. Effect of data perturbation

For evaluation of the proposed method, perturbation was produced by adding Gaussian noise to original data. To explore an effect of perturbation, additional evaluations were performed with other perturbation methods such as mixup [48] and frame-shift. For data mixup, $Mix_\lambda(a, b)$ defined in (2) was used to generate manipulated data for two original data a and b. A delay factor (i.e. the number of frame) for frame shifting was generated by Gaussian random process with zero-mean and 40-standard deviation. Then, a new data point was generated by truncating 10 second from the delay factor and padding with the rest of original data. The results are summarized in TABLE IV with the mean and standard deviation in 5 cross-validation repetitions.

As noted in the results, frame-shifting is the most effective technique tested. Adding Gaussian noise is also comparable to the frame-shifting (*p-value*=0.0799 and *p-value*=0.7215 in validation and public evaluation for classwise post-processing, respectively). On the other hand, the mixup method appears to yield worst results. One explanation is that this technique generates more overlapping events which may bias *Model II* to learn mapping of target sound features from these overlapping sounds. Naturally, another limitation is that the pseudo label estimation assumes a cap of 2 on the number of concurrent events, informed by the statistics of real test audios as shown in TABLE III.

TABLE IV
CLASS AVERAGING F-SCORE IN DIFFERENT PERTURBATIONS

perturbation	Validation	
	Global	Classwise
Adding noise	37.89 ± 1.01	39.76 ± 1.90
Mixup	34.35 ± 1.40	37.21 ± 0.79
Shifting	39.26 ± 1.77	41.61 ± 0.79
perturbation	Public evaluation	
	Global	Classwise
Adding noise	40.19 ± 1.59	42.81 ± 2.03
Mixup	36.15 ± 1.10	40.35 ± 0.90
Shifting	40.48 ± 1.53	43.26 ± 1.79

VI. DISCUSSION

Current state-of-the-art systems for sound event detection have been leveraging a combination of approach such as data augmentation, network architecture, semi-supervised learning, post-processing, and fusion in order to yield the best performance on the DCASE challenge. One of the best performing systems proposed by Miyazaki et al. explores all these aspects [29]. A self-attention model based on either transformer [7] or conformer [49] was used instead of the CRNN. And the self-attention model was trained based on MT approach. Although the system finally reached to 50.6% for class averaging f-score, it seems that the new model is comparable to the challenge baseline. When the self-attention model is trained without data augmentation and tested without both post-processing and fusion, the performance was reported to 28.6 % and 34.4 % for transformer and conformer, respectively. Although the f-scores were respectively improved to 41.0% and 41.7% by performing their classwise post-processing, it has an issue for a generalization of the parameters such threshold and filter length because both parameters were manually optimized on the ”validation” set.

Koh, et al. proposed a SCT loss and a deeper network based on modified CRNN, where a pooling is only performed on frequency domain for a high time-resolution, and feature-pyramid (FP) that leverages to discard unreliable predictions is incorporated with the CRNN [30]. In post-processing, the target sounds were grouped into two groups, background sounds and impulsive sounds, depending on the sound duration, then two different filters, whose length was heuristically decided, were applied for the sounds. According to the technical report, their implementation for ICT shows better performance than the ICT result in TABLE II. A possible reason is the use of ICT loss in combination with the MT loss. Also interpolation and time-frequency shifting were considered as a part of data augmentation from the loss function for SCT. The data augmentation in this manner seems be effective in improving the performance as shown in other submissions [29], [31], [50]–[52]. Since this study focuses on a method for semi-supervised learning, a simple way like adding noise is only considered to make manipulated data samples x' in Fig. 2.

Kim, et al. proposed another modified CRNN which is using more channels and skip-connected convolution layers with attention modules [31]. Instead of MT approach, the network was trained with a cross-entropy between model prediction and pseudo label to resolve the issue of MT approach in Fig. 1(b). The pseudo label was estimated on an weighted sum of model prediction and truth label with a preset weight. According to the technical report, it seems that this approach outperformed the results in TABLE II. However, it is unfair to directly compare to the numbers in the TABLE II because Kim's method used a different architecture in the number of channels and skip connection as well as data augmentation based on interpolation and time-frequency masking.

In this study, all other configurations such as perturbation (i.e. data augmentation) and network architecture were fixed because this study tackles to develop an advanced method for semi-supervised learning. The effect of other configuration on the SED performance will be investigated as a future work.

VII. Conclusion

Sound event detection aims to identify sounds of interest as sound category and time boundaries for each sound interval. Deep neural networks are powerful models for the sound event detection; however, they are faced with the challenge of data acquisition and curation in order to provide accurate estimate of events that can guide supervised training of these networks. The current paper explores self-training using cross-referencing as well as classwise post-processing in order to leverage the unlabeled training data. The cross-referencing self-training is composed of two separate models. In training, one model estimates a pseudo label for unlabeled data with the prediction by itself, then the other model refers the pseudo label to train itself vice versa. In this way, these two models are separately trained so that a self-biasing risk in self-referencing model could be resolved. Additionally, a post-processing composed of thresholding and smoothing is explored to find sound intervals from the model prediction. This paper introduces a threshold estimation based on Extreme Value Theory and filter length estimation with the statistics of model prediction. These proposed approaches are tested on sound event detection task described by the recent DCASE challenge and shown to result in improved performance compared to the state-of-the art in semi-supervised learning.

Appendix A
Network architecture of CRNN and Training parameters

The basic network, Convolutional Recurrent Neural Network (CRNN), consists of 7-convolutional blocks for CNN and 2-Bidirectional Gate Recurrent Unit (BGRU) for RNN. Each convolutional block was composed of 2D-convolution layer with $[3 \times 3]$ kernel and $[1 \times 1]$ stride, 2D batch normalization layer, activation layer, dropout layer, and average pooling layer. The number of kernels was set to [16, 32, 64, 128, 128, 128 ,128] per block. The 2D batch normalization was performed along to the output channels. Gated Linear Unit (GLU) in (5) was applied to the activation layer. The dropout parameter was set to 0.5. In the average pooing layer, the stride size was defined as [[2, 2], [2, 2], [1, 2], [1, 2], [1, 2], [1, 2], [1, 2]] per block. Note that the pooling was performed twice along to time axis while the pooling was performed in 7-times along to frequency axis. The number of nodes in BGRU was set to 128 as the number of kernels at the top of the previous CNN.

Each model on this structure was trained for 300 epochs with Adam optimizer. And the best model on a cross-validation set with global post-processing is kept. The remaining parameters defined for training are following: 24 batch size (6 strong labeled data, 12 unlabeled data, and 6 weakly labeled data), adaptive learning rate limited to 0.001 for maximum.

Acknowledgment

This work was supported by NIH U01AG058532, ONR N00014-17-1-2736, N00014-19-1-2689 and NSF 1734744.

References

[1] K. Ahmad and N. Conci, "How deep features have improved event recognition in multimedia: A survey," *ACM Trans. on Multimedia Compu., Comm. and App.*, vol. 15, no. 2, 2019.

[2] Y. Lavner, R. Cohen, D. Ruinskiy, and H. Ijzerman, "Baby cry detection in domestic environment using deep learning," in *IEEE International Conference on the Science of Electrical Engineering,*. IEEE, 2017.

[3] S. Park, Y. Lee, D. K. Han, and H. Ko, "Subspace projection cepstral coefficients for noise robust acoustic event recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, New Orleans, LA, USA, 2017, pp. 761–765.

[4] E. Cakir, G. Parascandolo, T. Heittola, H. Huttunen, and T. Virtanen, "Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 6, pp. 1291–1303, 6 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7933050/

[5] F. Vesperini, L. Gabrielli, E. Principi, and S. Squartini, "Polyphonic Sound Event Detection by Using Capsule Neural Networks," *IEEE J. Selected Topics in Signal Proc.*, vol. 13, no. 2, pp. 310–322, 2019.

[6] S. Kothinti, K. Imoto, D. Chakrabarty, G. Sell, S. Watanabe, and M. Elhilali, "Joint Acoustic and Class Inference for Weakly Supervised Sound Event Detection," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 5 2019, pp. 36–40.

[7] Q. Kong, Y. Xu, W. Wang, and M. D. Plumbley, "Sound Event Detection of Weakly Labelled Data with CNN-Transformer and Automatic Threshold Optimization," *IEEE/ACM Trans. Audio Speech and Lang. Proc.*, vol. 28, pp. 2450–2460, 2020.

[8] B. Shi, M. Sun, C.-c. Kao, V. Rozgic, S. Matsoukas, and C. Wang, "Semi-supervised Acoustic Event Detection Based on Tri-training," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 5 2019, pp. 750–754.

[9] L. Lin, X. Wang, H. Liu, and Y. Qian, "Guided Learning for Weakly-Labeled Semi-Supervised Sound Event Detection," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 5 2020, pp. 626–630.

[10] N. Turpault and R. Serizel, "Training Sound Event Detection On A Heterogeneous Dataset," in *DCASE workshop*, 2020.

[11] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *International Conference on Machine Learning Workshop*, 2013, pp. 1–6.

[12] C. Wei, K. Shen, Y. Chen, and T. Ma, "Theoretical analysis of self-training with deep networks on unlabeled data," in *International Conference on Learning Representations*, 2021.

[13] S. Park, A. Bellur, D. K. Han, and M. Elhilali, "Self-Training for Sound Event Detection in Audio Mixtures," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6 2021, pp. 341–345. [Online]. Available: https://ieeexplore.ieee.org/document/9414450/

[14] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[15] S. Mun, S. Shon, W. Kim, D. K. Han, and H. Ko, "Deep Neural Network based learning and transferring mid-level audio features for acoustic scene classification," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3 2017, pp. 796–800.

[16] 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.

[17] V. Sindhwani and S. S. Keerthi, "Large scale semi-supervixed linear SVMs," in *Proceedings of the Twenty-Ninth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, vol. 2006. New York, New York, USA: ACM Press, 2006, pp. 477–484.

[18] R. Dara, S. C. Kremer, and D. A. Stacey, "Clustering unlabeled data with SOMs improves classification of labeled real-world data," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, 2002, pp. 2237–2242.

[19] D. Erhan, A. Courville, Y. Bengio, and P. Vincent, "Why Does Unsupervised Pre-training Help Deep Learning?" *Proceedings of Machine Learning Research*, vol. 9, pp. 201–208, 2010.

[20] M. Sajjadi, M. Javanmardi, and T. Tasdizen, "Regularization with stochastic transformations and perturbations for deep semi-supervised learning," in *Advances in Neural Information Processing Systems*, no. Nips, 2016, pp. 1171–1179.

[21] S. Laine and T. Aila, "Temporal Ensembling for Semi-Supervised Learning," in *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, 10 2017, pp. 1–13.

[22] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems*, vol. 2017-Decem, no. Nips, 2017, pp. 1196–1205.

[23] V. Verma, A. Lamb, J. Kannala, Y. Bengio, and D. Lopez-Paz, "Interpolation Consistency Training for Semi-supervised Learning," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, vol. 2019-Augus. California: International Joint Conferences on Artificial Intelligence Organization, 8 2019, pp. 3635–3641.

[24] Y. Grandvalet and Y. Bengio, "Semi-supervised Learning by Entropy Minimization," in *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[25] ——, "Entropy Regularization," in *Semi-Supervised Learning*, 2006, pp. 29–46.

[26] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *AISTATS 2005 - Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, 2005, pp. 57–64.

[27] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-Supervised Self-Training of Object Detection Models," in *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*. IEEE, 1 2005, pp. 29–36.

[28] I. Dopido, J. Li, P. R. Marpu, A. Plaza, J. M. Bioucas Dias, and J. A. Benediktsson, "Semisupervised Self-Learning for Hyperspectral Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 7, pp. 4032–4044, 7 2013.

[29] K. Miyazaki, T. Komatsu, T. Hayashi, S. Watanabe, T. Toda, and K. Takeda, "Convolution-Augmented Transformer for Semi-Supervised Sound Event Detection," in *Detection and Classification of Acoustic Scenes and Events Workshop*, 2020, pp. 2–5.

[30] C.-Y. Koh, Y.-S. Chen, Y.-W. Liu, and M. R. Bai, "Sound Event Detection by Consistency Training and Pseudo-Labeling With Feature-Pyramid Convolutional Recurrent Neural Networks," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6 2021, pp. 376–380. [Online]. Available: https://ieeexplore.ieee.org/document/9414350/

[31] N. K. Kim and H. K. Kim, "Polyphonic Sound Event Detection Based on Residual Convolutional Recurrent Neural Network With Semi-Supervised Loss Function," *IEEE Access*, vol. 9, pp. 7564–7575, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9312148/

[32] ——, "Polyphonic Sound Event Detection Based on Residual Convolutional Recurrent Neural Network with Semi-Supervised Loss Function," *IEEE Access*, vol. 9, pp. 7564–7575, 2021.

[33] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk, and I. J. Goodfellow, "Realistic evaluation of semi-supervised learning algorithms," in *Advances in Neural Information Processing Systems*, 2018, pp. 3235–3246.

[34] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 27, 2019.

[35] S. Pouyanfar, Y. Tao, A. Mohan, H. Tian, A. S. Kaseb, K. Gauen, R. Dailey, S. Aghajanzadeh, Y. H. Lu, S. C. Chen, and M. L. Shyu, "Dynamic Sampling in Convolutional Neural Networks for Imbalanced Data Classification," in *IEEE Conference on Multimedia Inform. Proc. and Retrieval*, 2018, pp. 112–117.

[36] H. Lee, M. Park, and J. Kim, "Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning," in *2016 IEEE International Conference on Image Processing (ICIP)*, vol. 2016-Augus. IEEE, 9 2016, pp. 3713–3717.

[37] J. B. Broadwater and R. Chellappa, "Adaptive threshold estimation via extreme value theory," *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 490–500, 2010.

[38] M.-S. Yang, C.-Y. Lai, and C.-Y. Lin, "A robust EM clustering algorithm for Gaussian mixture models," *Pattern Recognition*, vol. 45, no. 11, pp. 3950–3961, 11 2012.

[39] R. Gencay and F. Selcuk, "EVIM: A software packages for Extreme Value Analysis in Matlab," in *Studies in Nonlienar Dynamics & Econometrics*, 2001, vol. 5, no. 3.

[40] S. Kang and J. Song, "Parameter and quantile estimation for the generalized Pareto distribution in peaks over threshold framework," *Journal of the Korean Statistical Society*, vol. 46, no. 4, pp. 487–501, 2017.

[41] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[42] N. Turpault, R. Serizel, J. Salamon, and A. P. Shah, "Sound Event Detection in Domestic Environments with Weakly Labeled Data and Soundscape Synthesis," in *DCASE Workshop*. New York University, 2019, pp. 253–257.

[43] G. Dekkers, S. Lauwereins, B. Thoen, M. W. Adhana, H. Brouckxon, B. V. D. Bergh, T. V. Waterschoot, B. Vanrumste, M. Verhelst, P. Karsmakers, and V. U. Brussel, "The SINS Database for Detection of Daily Activities in a Home Environment using an Acoustic Sensor Network," in *Detection and Classification of Acoustic Scenes and Events 2017*, no. November, 2017, pp. 1–5.

[44] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," pp. 2–5, 2015.

[45] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Proceedings of the 21st ACM international conference on Multimedia - MM '13*. New York, New York, USA: ACM Press, 2013, pp. 411–412.

[46] A. Mesaros, T. Heittola, and T. Virtanen, "Metrics for polyphonic sound event detection," *Applied Sciences (Switzerland)*, vol. 6, no. 6, 2016.

[47] B. Derrick, D. Toher, and P. White, "How to compare the means of two samples that include paired observations and independent observations: A companion to Derrick, Russ, Toher and White (2017)," *The Quantitative Methods for Psychology*, vol. 13, no. 2, pp. 120–126, 5 2017.

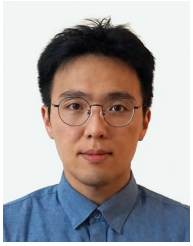
[48] H. Zhang;, M. Cisse;, Y. N. Dauphin;, and D. Lopez-Paz;, "Mixup: Beyond Empirical Risk Minimization," in *ICLR, International Conference on Learning Representations - Proceedings*, no. March, 2018, pp. 1–8.

[49] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Interspeech 2020*, vol. 2020-Octob. ISCA: ISCA, 10 2020, pp. 5036–5040.

[50] T. Yao, C. Shi, and H. Li, "Sound Event Detection In Domestic Environments Using Dense Recurrent Neural Network," in *Detection and Classification of Acoustic Scenes and Events Workshop*, 2020.

[51] Y. Liu, C. Chen, J. Kuang, and P. Zhang, "Semi-supervised Sound Event Detection Based on Mean Teacher with Power Pooling and Data Augmentation," in *Detection and Classification of Acoustic Scenes and Events Workshop*, no. Mil, 2020, pp. 4–6.

[52] Y. Huang, L. Lin, S. Ma, X. Wang, H. Liu, Y. Qian, M. Liu, and K. Ouchi, "Guided multi-branch learning systems for DCASE 2020 Task 4," in *Detection and Classification of Acoustic Scenes and Events Workshop*, no. November, 2020.



Sangwook Park received the B.S. degree in Electrical and Electronics Engineering from Chung-Ang University, Seoul, South Korea in 2012, and Ph.D. degree in electrical and computer engineering from Korea University, Seoul, South Korea in Aug. 2017. After a year of serving as Research Professor for audio signal analysis at Korea University, he was PostDoc fellow of Electrical and Computer Engineering in Johns Hopkins University from Sept. 2018 to Feb. 2022. Currently, he is assistant professor of Electronic Engineering in Gangneung-Wonju National University, South Korea. His current works involve acoustic signal analysis for detection and classification and building a computational model emulating mammalian auditory system.

David K. Han (Senior Member, IEEE) is the inaugural holder of the Bruce Eisenstein Endowed Chair and Professor of Electrical and Computer Engineering at Drexel University. He is an ASME Fellow and an IEEE senior member. He has previously held positions as research and faculty member at the Johns Hopkins University Applied Physics Laboratory, Army Research Laboratory, and the University of Maryland at College Park, and has additionally served as Distinguished IWS Chair Professor at the US Naval Academy. Han has authored or coauthored over 100 peer-reviewed papers, including four book chapters. His current research interests include computer vision, speech recognition,machine learning, and robotics.

Mounya Elhilali (Senior Member, IEEE) received her Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, MD, USA. She is a professor of Electrical and Computer Engineering with a secondary appointment at the department of Psychology and Brain Science at Johns Hopkins University. She directs the Laboratory for Computational Audio Perception, which examines human and machine hearing, with a focus on robust representation of sensory information in noisy soundscapes, problems of auditory scene analysis and cognitive control of auditory perception. She was named the Charles Renn faculty scholar and is the recipient of the National Science Foundation CAREER award and the Office of Naval Research Young Investigator award.