

JULY 26 2023

Using deep learning to track time × frequency whistle contours of toothed whales without human-annotated training data

Pu Li ; Xiaobai Liu ; Holger Klinck ; Pina Gruden; Marie A. Roch 



J. Acoust. Soc. Am. 154, 502–517 (2023)





<https://doi.org/10.1121/10.0020274>



LEARN MORE

Advance your science and career as a member of the
Acoustical Society of America

Using deep learning to track time \times frequency whistle contours of toothed whales without human-annotated training data

Pu Li,^{1,a)}  Xiaobai Liu,¹  Holger Klinck,^{2,b)}  Pina Gruden,³ and Marie A. Roch^{1,c)} 

¹Department of Computer Science, San Diego State University, San Diego, California 92182, USA

²K. Lisa Yang Center for Conservation Bioacoustics, Cornell Laboratory of Ornithology, Cornell University, New York 14850, USA

³Cooperative Institute for Marine and Atmospheric Research, Research Corporation of the University of Hawaii, Honolulu, Hawaii 96822, USA

ABSTRACT:

Many odontocetes produce whistles that feature characteristic contour shapes in spectrogram representations of their calls. Automatically extracting the time \times frequency tracks of whistle contours has numerous subsequent applications, including species classification, identification, and density estimation. Deep-learning-based methods, which train models using analyst-annotated whistles, offer a promising way to reliably extract whistle contours. However, the application of such methods can be limited by the significant amount of time and labor required for analyst annotation. To overcome this challenge, a technique that learns from automatically generated pseudo-labels has been developed. These annotations are less accurate than those generated by human analysts but more cost-effective to generate. It is shown that standard training methods do not learn effective models from these pseudo-labels. An improved loss function designed to compensate for pseudo-label error that significantly increases whistle extraction performance is introduced. The experiments show that the developed technique performs well when trained with pseudo-labels generated by two different algorithms. Models trained with the generated pseudo-labels can extract whistles with an *F1*-score (the harmonic mean of precision and recall) of 86.31% and 87.2% for the two sets of pseudo-labels that are considered. This performance is competitive with a model trained with 12 539 expert-annotated whistles (*F1*-score of 87.47%). © 2023 Acoustical Society of America. <https://doi.org/10.1121/10.0020274>

(Received 8 March 2023; revised 30 June 2023; accepted 6 July 2023; published online 26 July 2023)

[Editor: Shane Guan]

Pages: 502–517

I. INTRODUCTION

There are currently 72 recognized species of odontocetes (compared to only 14 baleen whale species) of which approximately two-thirds are known to produce whistles (Wursig and Perrin, 2009). Odontocete whistles are highly complex and variable communication signals and contain not only information about the species that produced the vocalization (Gillespie *et al.*, 2013; Jiang *et al.*, 2019) but also behavioral states (Taruski, 1979; Sjare and Smith, 1986), and, in some cases, individual identity (Caldwell and Caldwell, 1968; van Parijs and Corkeron, 2001; Janik *et al.*, 2013; Sayigh *et al.*, 2013; Kaplan *et al.*, 2014). Consequently, marine biologists frequently deploy hydrophones to study these marine mammals. Mid to high frequency signals, such as whistles and echolocation clicks (Oswald *et al.*, 2003), are frequently recorded at high sample rates (e.g., ≥ 192 kHz), resulting in extensive sound archives to be analyzed. Automated extraction (and subsequent species classification) of whistles from these data

remains a significant challenge in the field of animal bioacoustics, and new methods are needed to make the extraction process more efficient and reliable.

Most odontocete whistles feature characteristic contour shapes in the time-frequency (t-f) domain. Whistle extraction aims to determine the t-f bins of whistles in spectrograms, which then facilitates the subsequent tasks, e.g., classification of these acoustic signals to the species level. Although biologists can manually extract whistles as t-f contours in spectrograms, this task is highly labor intensive. To speed up the acoustic analysis process, various automated whistle extraction algorithms have been developed over the years (e.g., Mallawaarachchi *et al.*, 2008; White and Hadley, 2008; Mellinger *et al.*, 2011; Roch *et al.*, 2011; Gillespie *et al.*, 2013; Gruden and White, 2020; Li *et al.*, 2020; Wang *et al.*, 2021; Conant *et al.*, 2022).

Whistle extraction methods (e.g., Roch *et al.*, 2011) typically contain two steps. Most algorithms start by using peak detection algorithms to find regions of high energy that may belong to a whistle. In most cases, these are then examined to see if they are near other peaks and, therefore, likely to be parts of a whistle. This may be performed using deterministic (e.g., Mellinger *et al.*, 2011) or probabilistic (e.g., Gruden and White, 2020) trajectory models. These sets of peaks may be subjected to additional processing but are eventually reported as a whistle contour.

^{a)}Also at: Department of Computer Science, University of California Irvine, Irvine, CA 92697, USA.

^{b)}Also at: Marine Mammal Institute, Department of Fisheries, Wildlife, and Conservation Sciences, Oregon State University, Newport, OR 97365, USA.

^{c)}Electronic mail: marie.roch@sdsu.edu

More recently, deep-learning-based methods have been applied to whistle extraction. Li *et al.* (2020) trained convolutional neural networks (CNNs) to find candidate t-f bins of whistles in spectrograms. The CNN model outputs a confidence map for the spectrogram, where each t-f bin has a confidence score of whether this node contains part of a whistle signal. t-f nodes with confidence scores above a predefined threshold are connected into whistle contours using a graph search method (Roch *et al.*, 2011). Compared to using spectral peaks to extract whistles, the deep-learning-based method improved the *F1*-score by around 20% on a two-species (*Delphinus capensis* and *Tursiops truncatus*) benchmark dataset. However, training the model used thousands of manually annotated whistles, and analyst annotations were produced over several months.

To facilitate the application of deep-learning-based methods in situations where large, annotated datasets are unavailable, we explore ways to train the model with pseudo-labels. Pseudo-labels are approximative labels of whistles consisting of sequences of time \times frequency coordinates that trace the whistle in a spectrogram representation. In this work, we generated these by using previously published whistle extractors that require no or very little analyst training data. We do not expect the pseudo-labels to be as accurate as those produced by human analysts. We expect that some whistles will not be labeled, spurious labels may be generated, and errors in time and frequency labels may occur. Our setting does not require human analysts to annotate whistles nor to validate the generated pseudo-labels. Training deep neural networks (DNNs) with pseudo-labels is challenging due to the increased errors in the pseudo-labels as compared to analyst generated ones. Neural networks learn by adjusting network parameters to minimize a loss function that measures the difference between predictions and expected labels. Consequently, when a DNN is trained to fit these less accurate pseudo-labels well, the model will typically result in unsatisfactory performance.

Errors in the pseudo-labels can be thought of as a form of noise in the label set. The machine learning community has proposed three categories of modified loss functions to improve model robustness to label noise (Song *et al.*, 2022), including using distance metrics robust to label noise, reweighting samples according to their label quality, and loss correction with noise estimation.

First, researchers developed novel distance metrics in loss functions. Ghosh *et al.* (2017) showed that symmetric loss functions, e.g., mean absolute error (MAE), led to a smaller performance drop compared to nonsymmetric loss functions, e.g., categorical cross-entropy (CCE), when there are noisy labels. Zhang and Sabuncu (2018) found that MAE may perform poorly with DNNs and proposed to use negative Box-Cox transformation as a noise robust loss function, which surpassed MAE and CCE on varying label noise scenarios. Wang *et al.* (2019b) added reversed cross-entropy to the original cross-entropy loss, which formed a symmetric cross-entropy loss and reduced model overfitting

to noisy labels. Ma *et al.* (2020) normalized loss functions by dividing the sum of loss among all possible labels, but the resultant model tended to underfit. To address this problem, they further proposed active passive loss that combined normalized loss of two types: one only optimized on the label class (active loss) and one optimized on all classes (passive loss). Kim *et al.* (2019) and Kim *et al.* (2021) proposed negative learning (NL) loss to deal with noisy label on an image classification task. Compared to classical loss functions, which encourage neural network models to maximize the label class, NL minimizes the scores of the complementary non-label classes. Assuming that label noise is not too great, the NL loss tends to reduce the impact of a mislabeled example.

Second, samples may be weighted differently in the loss function. Natarajan *et al.* (2013) assumed the existence of class-dependent label noise on a binary classification dataset, and they modified the original loss to a weighted surrogate loss according to manually assigned noise rates and sample labels. Wang *et al.* (2019a) calculated the gradients of the training loss with regard to the logit vector and improved MAE by giving samples different weights according to the magnitude of gradients. Su *et al.* (2021) applied the annotator robust loss to edge detection data that are annotated by multiple annotators. If annotators cannot agree on the label of one pixel, this pixel was removed from the loss calculation. As the number of edge pixels is usually much smaller than that of background pixels, the annotator robust loss balanced their contribution by using different weight factors for the two classes.

Finally, the noise distribution may be estimated to correct the loss function. Goldberger and Ben-Reuven (2017) viewed the correct label as a latent random variable and modeled the noise by an additional softmax layer, which predicted the probability of correct hidden labels. Patrini *et al.* (2017) combined noise rate estimation algorithms and DNNs, where the estimated transition matrix corrected the loss function to make it equal to the original loss computed on clean labels. Tanno *et al.* (2019) modeled the annotation errors of each annotator with a confusion matrix, and they added a regularization term that jointly optimized the confusion matrix and model predictions. Xia *et al.* (2019) trained the classifier with noisy labels and initialized the label transition matrix based on their classifier's predictions and then retrained the model with a learnable variable that automatically revised the transition matrix.

All of the above methods work well on their target image classification tasks but translating these methods to dual-class (noise and whistle energy) t-f node predictions within audio requires the algorithms to be adapted to a new two-class domain that leverages the local context of surrounding t-f nodes to make correct predictions. In some cases, e.g., careful manual estimation of noise distributions, the methods are not well-aligned with the goal of this study, which is to perform predictions without the need for extensive analysis of the pseudo-labels. Direct application of these methods to a whistle extraction task did not seem

guaranteed to be a fruitful direction and we, therefore, designed a method inspired by these techniques. To the best of our knowledge, there is no previous method that uses pseudo-label training for the whistle extraction task. Specifically, we propose a method to re-weight different components in the loss function, which will reduce the effect of incorrect labels. We observe that our pseudo-labels frequently miss whistle signals. When models are directly trained with a loss function that assigns the same weight to each t-f bin, such models may prefer to predict whistles as background noise due to the presence of mislabeled whistle energy within the set of pseudo-labels. Consequently, we observe a tendency for the network to predict whistles with high precision but low recall.

To encourage the network to predict more whistles, we divide the t-f bins into two categories: foreground bins, where a pseudo-label indicates that whistle energy occurs, and background bins otherwise. We add a regularization term to re-weight foreground and background t-f bins in the loss function. The modified loss encourages the model to make correct predictions under expected label noise, e.g., when the pseudo-label missed part of a whistle. However, pseudo-labels may contain multiple types of errors, and although the modified loss function may suppress one type of error, it may encourage the model to have another type of error. For example, improving the weight of foreground t-f bins may help reduce false negative predictions but it also increases the chance of false positive predictions. Inspired by the work of focal loss (Lin et al., 2017), we add a multiplication factor in the regularization term so that it dynamically adjusts the weight according to the prediction and pseudo-label to reduce the undesired errors.

To examine our method and eliminate the need for manual annotations, we applied the proposed method to train a CNN-based whistle extractor on two different sets of pseudo-labels, generated by two different whistle extraction algorithms selected from the existing literature.

II. METHODS

A. Dataset

We used the acoustic data from the Detection, Classification, Localization, and Density Estimation (DCLDE) workshop (DCLDE Organizing Committee, 2011) for model training and evaluation. This dataset consists of approximately 32 h of recordings collected for five species of odontocetes: bottlenose dolphins (*T. truncatus*), long- and short-beaked common dolphins (*D. capensis*, *Delphinus delphis*), melon-headed whales (*Peponocephala electra*), and spinner dolphins (*Stenella longirostris*). Two types of hydrophones were deployed, ITC 1042 (International Transducer Corp., Santa Barbara, CA) and HS 150 (Sonar Research and Development Ltd., Beverly, UK) hydrophones, for collecting the data. The hydrophones were towed by the R/V David Starr Jordan, mounted to the stationary platform R/P FLIP (Fisher and Spiess, 1963), and deployed from small boats. The deployment depths of the hydrophones were 10–30 m.

The acoustic signals were sampled at 192 kHz with 16- or 24 bit quantization.

1. Data preparation

As in our previous work (Li et al., 2020), we transformed the acoustic data into log-magnitude spectrograms before using them to generate pseudo-labels or as input to a trained whistle extraction model. Discrete Fourier transforms (DFTs) were performed on 8 ms Hamming-windowed frames (125 Hz bandwidth) every 2 ms. These parameters were empirically set in Roch et al. (2011) as a trade-off between frequency and time resolution. Longer analysis windows with better frequency resolution tend to blur rapidly changing whistle signals. Examples of whistle signals from the five species are shown in Fig. 9 in the Appendix. We empirically restricted the log10-magnitude spectrogram to the range [0,6], clamping values to a range of 0–6. This corresponds to an uncalibrated intensity range of 0–120 dB, which was then normalized to the range [0,1]. We limited the spectrogram to the frequency range of 5–50 kHz (361 frequency bins), which covered most delphinid whistles and their harmonics. The spectrograms were divided into 3-s long nonoverlapping segments for model training and evaluation. The spectrogram segments from training datasets were further divided into patches of size 64 (128 ms) × 64 (8 kHz) before model training.

2. Training dataset

We used two nonoverlapping subsets of the DCLDE data for model training. First, we used a labeled subset for our supervised training experiments. The DCLDE dataset provides detailed t-f annotations of whistles for 45 recordings with a total duration of approximately 3 h. Annotations were previously produced for these data using an interactive software tool that let expert analysts trace whistles by placing control knots of cubic splines, the details of which can be found in Roch et al. (2011). Each recording contains a single species based on visual identification. Among these annotated recordings, we chose 30 recordings that were not used for evaluation in Roch et al. (2011) as our “labeled dataset.” These audio files recorded 127 min of odontocete calls and included 12 539 annotated whistles. Pseudo-label experiments used a larger unlabeled subset of the DCLDE data. These data consist of 348 recordings, and the total duration is around 29 h. This set of data is referred to as our “unlabeled dataset.”

3. Evaluation dataset

We used a subset of annotated acoustic data from the DCLDE workshop 2011 for evaluation. This subset consists of 12 audio files for bottlenose dolphins, long-beaked common dolphins, melon-headed whales, and spinner dolphins. The total duration of those recordings was around 43 min, and the t-f coordinates of 6011 whistles were annotated by analysts. All of these files were used for evaluation in the work of Roch et al. (2011). We did not use the recordings of

TABLE I. Summary of the number of whistles and recording duration per species in the evaluation dataset.

Species	Number of whistles	Duration (s)
Bottlenose dolphin	354	652.2
Long-beaked common dolphin	557	833.9
Melon-headed whale	338	680.7
Spinner dolphin	686	425.9

short-beaked common dolphins because of some annotation errors. The details of the number of annotated whistles per species that we expected to retrieve are summarized in Table I. Criteria for which whistles were expected to be retrieved is detailed in the metrics section (Sec. II E), and the specific files are summarized in Table III in the Appendix.

B. Pseudo-label generation

We use the spectral peak detection and graph search algorithm implemented in *Silbido*¹ (Roch *et al.*, 2011) to extract whistles from the unlabeled dataset. The spectral peak detection algorithm smooths the spectrograms with a median filter over each 3×3 t-f grid. Then it subtracts the mean value over a 3-s window in each frequency bin. If one t-f bin has a signal-to-noise ratio (SNR) larger than 10 dB and no other bins within ± 250 Hz have a larger magnitude than this t-f bin, it is a spectral peak. Next, the graph search algorithm manages the candidate detections with sets of graphs. Each graph depicts one or more candidate whistle contours where a sequence of spectral peaks is connected. Each spectral peak either starts a new graph or is added to existing graphs. Peaks are added to existing graphs if they are a good fit to adaptive polynomial predictions of graph trajectories and, otherwise, used to seed new graphs. Polynomial order is driven by goodness of fit as measured by the adjusted R^2 coefficient (Dillon and Goldstein, 1984), and spectral peaks are merged into an existing graph when they are within 50 ms of the last end point in the graph and 1000 Hz of the fitted polynomial curve. Graph state is maintained across 3-s blocks, permitting graphs to represent spectral peaks from whistles that cross processing blocks. Once a graph is no longer eligible to incorporate additional spectral peaks, whistles are extracted from the graph. When interior nodes have more than a pair of edges, the rate of change on both sides are examined to determine if multiple whistles crossed the interior node. We remove detected whistles that are shorter than 150 ms as per Roch *et al.* (2011).

To further examine our method on pseudo-labels with different characteristics, we used the sequential Monte Carlo probability hypothesis density (SMC-PHD) whistle extractor² by Gruden and White (2020) to generate the second set of pseudo-labels. Briefly, the algorithm uses computationally tractable approximation of the multi-target Bayes filter to track whistle contours based on spectral peaks from preprocessed spectrograms. Preprocessing of spectrograms is

based on established methods (Gillespie *et al.*, 2013; Gruden and White, 2016) to reduce noise and interfering signals. If t-f bins have magnitudes larger than 8 dB on the normalized spectrogram and are within the frequency range of 2–50 kHz, they are considered to be spectral peaks. These peaks are used as measurements for the SMC-PHD algorithm to track whistles. The SMC-PHD filter is a recursive filter that propagates the first-order moment of the multi-target posterior [called probability hypothesis density (PHD)] in time through prediction and update steps. The PHD function at each time step is approximated by a cloud of weighted particles. Particle locations and weights are predicted and updated according to the sequential Monte Carlo principles and PHD equations, respectively. The SMC-PHD implementation of Gruden and White (2020) used in this work employs a trained radial basis function (RBF) network to estimate the particle locations in the prediction step. The training data consist of 3 min of recording and 185 annotated whistles, and these data are not included in our training or evaluation dataset. Consequently, this second algorithm requires some analyst-annotated training data, but it is trivial when compared to the requirements for deep learning algorithms. New whistles are introduced to the filter through a birth model that incorporates measurements and priors based on training data. Additionally, the filter incorporates false alarms and missed detections in the problem formulation. At each time step, whistle states (representing whistle contour peaks) are estimated and their identities tracked based on labeled particles as outlined in Gruden and White (2020).

Irrespective of the whistle extraction algorithm, we generate bin-wise pseudo-labels for each 3-s spectrogram segment. The pseudo-label is initialized as a zero matrix of the same size as the spectrogram segment. We draw the whistle contour on the matrix with the `cv2.polyline()` method in the Python OpenCV library (Bradski, 2000). The thickness of the polyline is empirically set to two. The pseudo-labels have element values normalized to values between zero (background) and one (whistle), respectively. Similar to the training spectrograms described in Sec. II A 1, the pseudo-labels are divided into 64×64 patches that match the spectrogram patches for model training. If the pseudo-label marks at least one t-f bin in the patch as containing whistle energies, we consider this patch to be a “positive patch.” Otherwise, the patch is considered to be a “negative patch.” As there are many more negative patches than positive patches, we balance the training dataset by randomly selecting the same number of negative patches as positive patches for model training.

C. CNN-based whistle extraction

We use the deep whistle contour (DWC) detector³ implemented by Li *et al.* (2020) as our model for whistle extraction (Fig. 1). First, a CNN model, the whistle extraction network, takes a spectrogram as input and predicts a confidence map of the same size as the input spectrogram.

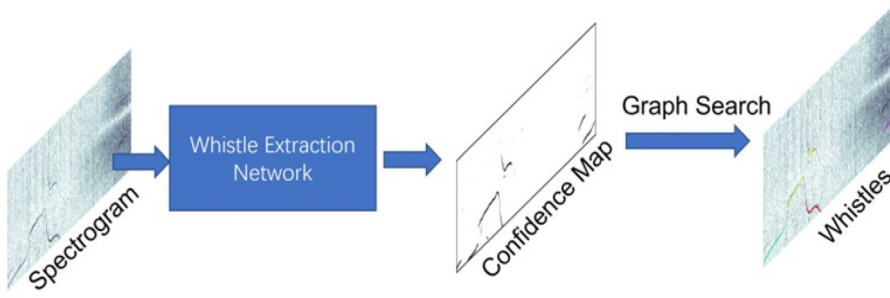


FIG. 1. (Color online) Illustration of the whistle extraction algorithm from Li *et al.* (2020). The neural network identifies whistle energy from input spectrograms and is processed by a subsequent algorithm to extract whistle annotations.

The confidence within each t-f bin indicates the probability that this bin contains whistles. The confidence map is used to predict peaks for a modified version of the graph search method that was summarized in Sec. II B. Confidence map t-f bins are labeled as peaks when the probability of attribution to whistle energy is larger than 0.5 and the bin contains a local maximum along the frequency axis. Whistle contours are produced from the set of peaks using a modified version of the graph search method summarized in Sec. II B.

Complete details of the network may be found in Li *et al.* (2020), but to summarize briefly, the network consists of ten convolutional layers. The inner eight convolutions consist of four residual network blocks (He *et al.*, 2016) each followed by batch normalization (Ioffe and Szegedy, 2015) with a parametric rectified linear unit activation function (He *et al.*, 2015). Denoting y and \hat{y} as the vectorized label and CNN output, respectively, the training loss function is

$$L_{\text{base}}(\hat{y}, y) = \sqrt{\|y - \hat{y}\|_2^2 + \varepsilon^2}, \quad (1)$$

where ε is a small constant (1×10^{-3}), and $\|y - \hat{y}\|_2$ is the L2-norm of the difference between y and \hat{y} . This baseline loss function encourages the CNN model to predict the value of the pseudo-label. We train the model for 1×10^6 iterations (around 88 epochs). The learning rate is initially 0.001 and multiplied by 0.1 every 400 000 iterations. The other training hyperparameters and graph search parameters are the same as those in the implementation of Li *et al.* (2020). Tuning of these hyperparameters is beyond the scope of this paper, but we empirically found that this set of parameters worked well in our experiments.

D. Pseudo-label learning

Let us consider the errors in the pseudo-labels. Figure 2 shows two typical examples of whistles extracted by graph search. The extracted whistles typically have high bin-wise precision but low bin-wise recall, i.e., the extracted contours mostly cover t-f bins that have whistles, but there are a significant number of whistle t-f bins missed.

We use a synthetic toy example (Fig. 3) to illustrate the impact of whistles that are missed by the pseudo-label generator. In this case, the true label contains two whistles, and the pseudo-label generator missed one of them. As these whistles have similar appearance, our whistle extraction

model may tend to make the same prediction for both whistles. Therefore, there are two likely predictions: the CNN model recognizes both whistles (prediction 1) or misses all of them (prediction 2). These two predictions have the same loss value under L_{base} , which means that the model may choose either one of the predictions during training. To encourage the model to have the correct prediction, we modify the loss function in Eq. (1) by adding a regularization term such that

$$L_{\text{recall}}(\hat{y}, y) = L_{\text{base}}(\hat{y}, y) + \lambda \sqrt{\|(\hat{y} - y)y\|_2^2 + \varepsilon^2}, \quad (2)$$

where $\lambda \in \mathbb{R}^+$ is a constant number, and ε is 1×10^{-3} . By modifying the training objective, we increase the penalty for the model missing t-f bins that pseudo-labels have marked as containing whistle energy. When $\lambda > 0$, prediction 1 has a lower loss than prediction 2, i.e., the model will prefer prediction 1 during training. Therefore, the modified loss function will help the model to detect whistles missed in pseudo-label and increase prediction recall.

However, the above conclusion may not apply to the case where pseudo-labels incorrectly predict whistles in areas of background noise or confounding signals (Fig. 4). Similar to the previous case, inaccurate pseudo-labels are driving the model to associate examples of whistles and background noise/confounding signals as the same category. This can result in the model predicting noise or confounding signals as whistles (prediction 1), or other training examples may result in the model correctly recognizing the inaccurate pseudo-label as background (prediction 2). L_{recall} [Eq. (2)] will make the model prefer prediction 1 instead of prediction 2, which leads to an increased false positive rate. To mitigate this problem, we modify the loss function to be

$$L_{\text{recall}}(\hat{y}, y) = L_{\text{base}}(\hat{y}, y) + \lambda(1 - R_{\text{soft}}(\hat{y}, y))^\gamma \times \sqrt{\|(\hat{y} - y)y\|_2^2 + \varepsilon^2}, \quad (3)$$

where $\lambda \in \mathbb{R}^+$ and $\gamma \in \mathbb{N}^+$ are constant parameters, and ε is 1×10^{-3} . $R_{\text{soft}}(\hat{y}, y)$ is the soft recall of prediction \hat{y} to pseudo-label y ,

$$R_{\text{soft}}(\hat{y}, y) = \frac{\|y\hat{y}\|_1}{\|y\|_1 + \varepsilon}, \quad (4)$$

which measures the rate at which t-f bins marked as whistles in the pseudo-label are detected. ε is a small positive

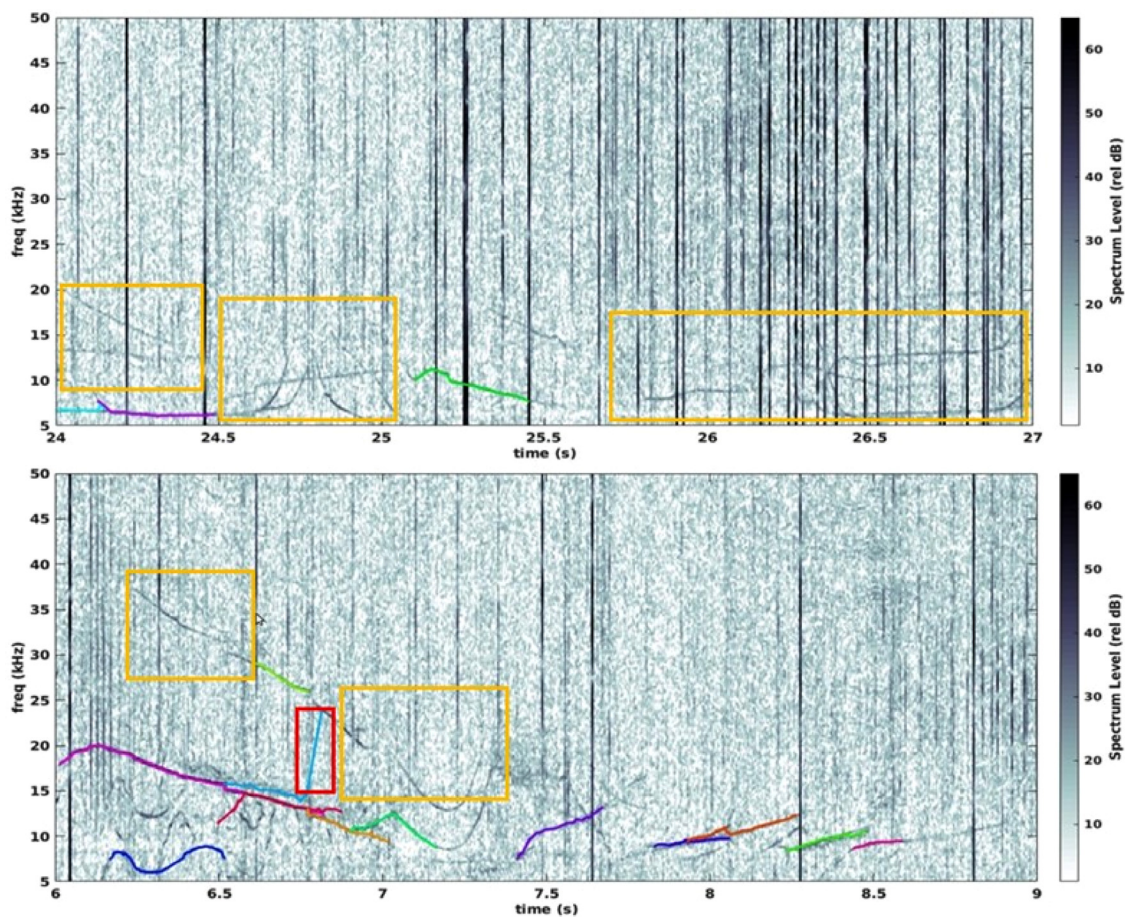


FIG. 2. (Color online) Two examples of the whistles detected by graph search (Roch *et al.*, 2011). The extracted whistles are shown as colored polylines. The contrast of the spectrogram is improved for better visualization. We highlight examples of the missed whistles and false positive detections with orange and red bounding boxes, respectively.

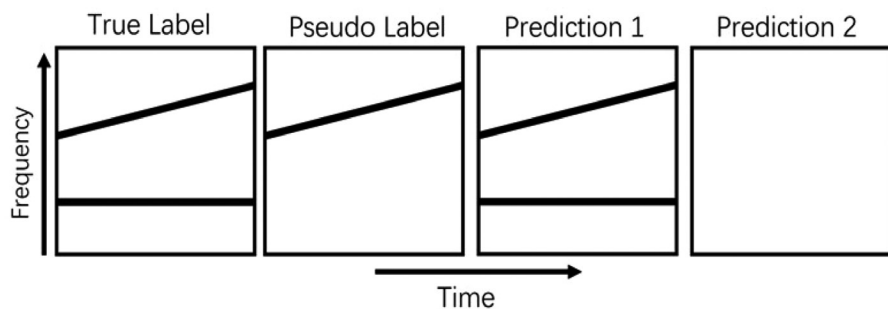


FIG. 3. A toy example for the case when pseudo-labels do not include some of whistles in ground truth label.

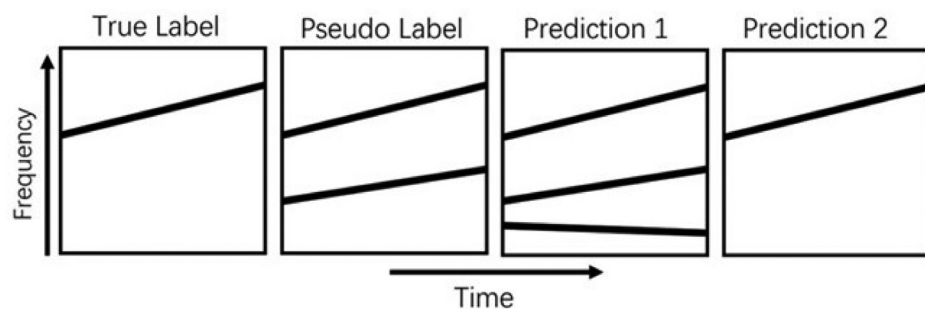


FIG. 4. A toy example for the case when we have false positive detections in pseudo-labels.

constant (1×10^{-5}). The $(1 - R_{\text{soft}}(\hat{y}, y))^{\gamma}$ term increases the penalty $\lambda \|(\hat{y} - y)\|_2$ when t-f nodes pseudo-labeled as whistles are predicted with low confidence. Compared to the first proposed penalized loss [Eq. (2)], the updated regularization term leads to lower penalties when the model predicts higher scores on t-f nodes with whistle pseudo-labels. For example, if we have $\gamma = 1$, the penalty weights become zero and 0.5λ for predictions 1 and 2 in Fig. 4, respectively. At the same time, the penalty weights are zero and λ for predictions 1 and 2 in Fig. 3, respectively. While we have the same penalty as in Eq. (2) for the wrong prediction (prediction 2) in Fig. 3, we reduce the penalty for the model to have the correct prediction (prediction 2) in Fig. 4. Therefore, the model is less encouraged to recognize background noise as whistles with Eq. (3). The parameter γ may adjust the influence of recall. We note that Eq. (3) is a generalization of Eqs. (1) and (2). If $\gamma = 0$, Eq. (3) is the same as Eq. (2). If γ is infinitely large and the recall is below one, Eq. (3) is the same as Eq. (1).

Equation (3) helps model training when the pseudo-label contains more false negatives than false positive detections. When the opposite is true and false positives are more prevalent in the pseudo-labels, we can revise the training objective to address this such that

$$L_{\text{prec}}(\hat{y}, y) = L_{\text{base}}(\hat{y}, y) + \lambda(1 - P_{\text{soft}}(\hat{y}, y))^{\gamma} \times \sqrt{\|(\hat{y} - y)(1 - y)\|_2^2 + \varepsilon^2}, \quad (5)$$

where $\lambda \in \mathbb{R}^+$ and $\gamma \in \mathbb{N}^+$ are constant parameters, ε is 1×10^{-3} , and $P_{\text{soft}}(\hat{y}, y)$ is the soft precision of prediction \hat{y} to pseudo-label y ,

$$P_{\text{soft}}(\hat{y}, y) = \frac{\|y\hat{y}\|_1}{\|\hat{y}\|_1 + \epsilon}, \quad (6)$$

where ϵ is a small positive constant number (1×10^{-5}). L_{prec} will encourage the model to generate prediction 2 in the case of Fig. 4, reducing false positives. In pseudo-label training, the effectiveness of L_{recall} or L_{prec} is likely to depend on the pseudo-label noise statistics. If the pseudo-labels include more false positives, L_{prec} will likely be the better choice. Conversely, if the pseudo-labels include more false negatives, L_{recall} should be chosen for model training.

E. Metrics

We evaluate the model performance using the evaluation code in *Silbido*. The evaluation starts with a matching process between detections and ground truth labels. If detected and annotated whistles overlap in time for more than 30% of the annotated whistle and the mean frequency difference is less than 350 Hz, this pair of detected and annotated whistles are considered a match. As in Roch *et al.* (2011) and Li *et al.* (2020), we only consider the analyst-annotated whistles with a duration ≥ 150 ms and a SNR ≥ 10 dB over at least one-third of the whistle. Any annotated whistles that did not meet these criteria were omitted from the analysis. Detections that

matched discarded ground truth annotations were neither counted toward nor against performance.

After matching, *Silbido* provides five metrics for whistle extraction performance. Precision indicates the percentage of correctly detected ground truth whistles. Recall is the percentage of ground truth whistles missed. The next three metrics are designed to measure the quality of detections. Deviation shows the average frequency deviation of the detected whistle to matched annotation. For annotated whistles that are matched to detections, coverage is the percentage of these whistles' durations that was detected. Finally, the fragmentation metric calculates the average number of detections matched to the same ground truth whistle and provides an indication of how often a whistle is split into multiple segments in the detection process. Ideally, one would have detections with zero deviation, 100% coverage, and a fragmentation score of one.

We calculate the *F1*-score, the harmonic mean of precision and recall, as an overall metric of the extraction performance. We evaluate our model on each species independently and report the performance averaged on different species.

III. EXPERIMENTS

A. Pseudo-label generated by graph search

We designed a series of experiments to validate our proposed methods. In a first step, we extracted whistles with spectral peaks detection and graph search method in *Silbido*, and this experiment is referred to as "graph search." Next, we trained two models with the L_{base} loss function [Eq. (1)]. The first of these models used the analyst annotations and is referred to as " $L_{\text{base-annotation}}$." The second model, " $L_{\text{base-graph}}$," used the same loss function but was trained with *Silbido* graph search generated pseudo-labels from the larger unlabeled dataset.

To examine the effectiveness of the proposed regularization penalties L_{recall} and L_{prec} , we trained additional models on the unannotated data using graph search labels. Experiments using these models are denoted " $L_{\text{recall-graph}}$ " and " $L_{\text{prec-graph}}$." Various values of λ and γ were empirically explored to find the optimal parameter setting on our dataset. Specifically, we use values of 0, 1, 2, or 4 for the exponent parameter γ in L_{recall} and L_{prec} . For each fixed value of γ , we explored varied λ values until we find a peak *F1*-score. In the L_{recall} experiments, we used $\lambda \in \{0.5, 1, 2, 3, 4\}$, $\{2, 4, 6, 8\}$, $\{4, 6, 8, 10\}$, and $\{4, 6, 8, 10, 15, 20, 25, 30\}$ for $\gamma = 0, 1, 2$, and 4, respectively. We explored larger values of λ when γ is larger in L_{recall} because larger γ led to lower weight in the regularization term. We used the same set of λ in L_{prec} , $\lambda \in \{0.01, 0.1\}$, as we observed that larger λ resulted in lower *F1*-scores and experiments with $\lambda = 0$ (removal of the regularization term) had the best *F1*-scores.

B. Pseudo-labels generated by SMC-PHD

To further validate our proposed method, we substituted an alternative whistle extraction method to generate a different set of pseudo-labels. We used SMC-PHD

TABLE II. Summary of various methods' whistle extraction performances. Scores indicating the harmonic mean ($F1$) of precision and recall, the mean deviation in frequency from analyst annotations (μ_σ), the percentage of each whistle that was detected (coverage), and the mean number of connected segments for each whistle (fragmentation).

Method	$F1$ (%)	Precision (%)	Recall (%)	μ_σ (Hz)	Coverage (%)	Fragmentation (detected segments/whistle)
L_{base} -annotation	87.47	89.50	85.93	92.00	88.08	1.13
Graph search	75.95	81.13	72.28	101.00	81.05	1.23
SMCPHD	83.40	76.55	92.45	108.00	70.93	1.80
$\text{SMCPHD} \geq 150$ ms	74.38	95.85	60.88	103.50	72.88	1.23
L_{base} -graph	74.14	94.15	61.53	144.75	77.50	1.18
L_{base} -SMCPHD	82.34	94.98	72.75	122.75	81.75	1.18
L_{base} -SMCPHD ≥ 150 ms	70.97	98.45	56.08	119.75	73.70	1.20
L_{recall} -graph	86.31	89.55	83.33	154.25	86.73	1.18
L_{recall} -SMCPHD	86.42	87.18	85.85	134.25	87.30	1.15
L_{recall} -SMCPHD ≥ 150 ms	87.20	88.78	85.78	134.75	87.30	1.18

(Gruden and White, 2020) to extract whistles, and the extraction result was referred to as “SMCPHD.” We used the RBF motion model, which requires a modest amount of analyst-annotated training data, with Gruden and White using a small training set of 185 whistles from several minutes of annotated data that do not overlap with our test data. Consequently, this method is not entirely free of analyst annotations. As our method and graph search discarded detections that were shorter than 150 ms, we created a second set of pseudo-labels where only detections that were at least 150 ms were retained: “SMCPHD ≥ 150 ms.”

We trained the whistle extraction model with the L_{base} loss function on these two sets of pseudo-labels, which are hereafter referred to as “ L_{base} -SMCPHD” and “ L_{base} -SMCPHD ≥ 150 ms.” As SMC-PHD exhibits the same characteristics as the graph search of tending to produce more false negatives than false positives, we trained models using the

L_{recall} loss function and varied the values of γ and λ on these two sets of pseudo-labels, which were referred to as “ L_{recall} -SMCPHD” and “ L_{recall} -SMCPHD ≥ 150 ms,” respectively. We use $\gamma = 0, 1$ for L_{recall} . Specifically, we used $\lambda \in \{1, 2, 3, 4, 5, 6\}$ for $\gamma = 0$ and $\lambda \in \{2, 4, 6, 8, 10, 12, 11, 12, 14, 16\}$ for $\gamma = 1$ for L_{recall} -SMCPHD ≥ 150 ms. For the experiment that did not discard short detections, L_{recall} -SMCPHD, we used $\lambda \in \{1, 2, 3, 4\}$ for $\gamma = 0$ and $\lambda \in \{2, 3, 4, 5, 6, 8, 10\}$ for $\gamma = 1$. We did not execute experiments using the L_{prec} loss function as SMC-PHD exhibits similar patterns of error in the pseudo-labels.

IV. RESULTS

A. Comparison between the proposed method and baselines

We summarize the performance of our method and several baselines in Table II. There are three types of baselines:

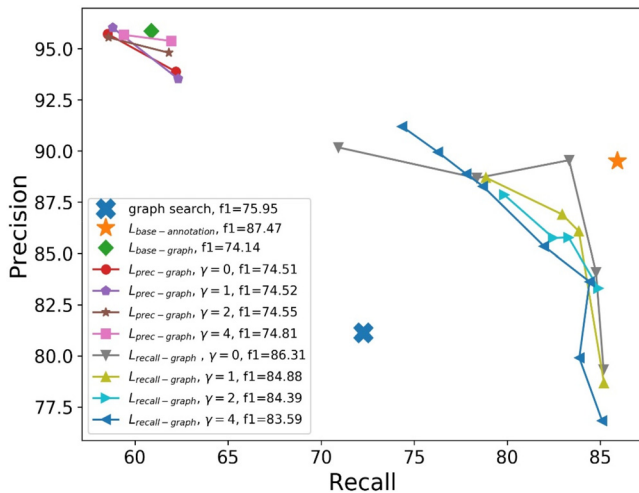


FIG. 5. (Color online) Summary of model performances derived from our graph search experiments. Systems trained with pseudo-labels are compared to the target performance of a system trained with human analyst-labeled data (\star), the algorithm that generated the pseudo-labels (\times), and a baseline loss function (\diamond). Each color curve shows the system performance when models are trained with L_{recall} or L_{prec} under a fixed γ and varied λ . The best $F1$ -score among the experiments in each curve is shown in the legend. Smaller values of λ result in lower recall for L_{recall} curves and lower precision for L_{prec} .

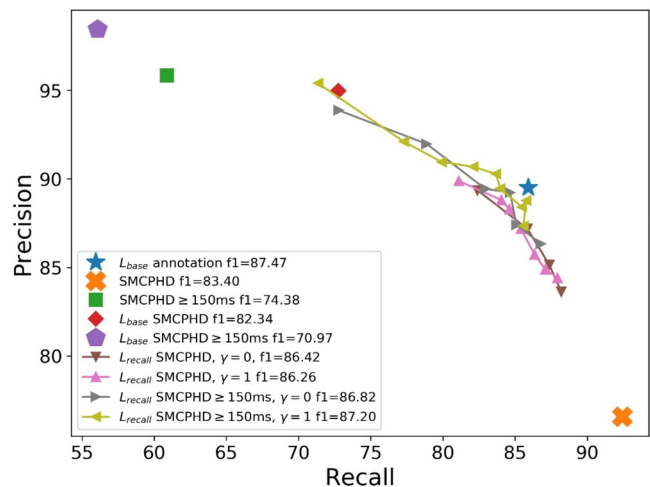


FIG. 6. (Color online) Summary of model performances derived from our experiments using SMC-PHD generated pseudo-labels. Systems trained with pseudo-labels are compared to the target performance of a system trained with human analyst-labeled data (\star), the algorithms that generated the pseudo-labels (\times , \blacksquare), and networks trained with a baseline loss function (\diamond , \blacklozenge). The color curves show the system performance when models are trained with L_{recall} under a fixed γ and varied λ . The best $F1$ -score among the experiments in one curve is shown in the legend. For comparison, a curve corresponding to graph search generated labels (L_{recall} graph) is also shown.

(i) the model trained with analysts' annotation, (ii) the performance of the algorithms used to produce pseudo-labels, and (iii) the model trained with pseudo-label and L_{base} . We also show the best performance using L_{recall} and pseudo-

labels. Experiments using the L_{prec} loss are included in the ablation study of Sec. IV B.

The network model trained using analyst annotations and the baseline loss function, " $L_{\text{base-annotation}}$," results in a

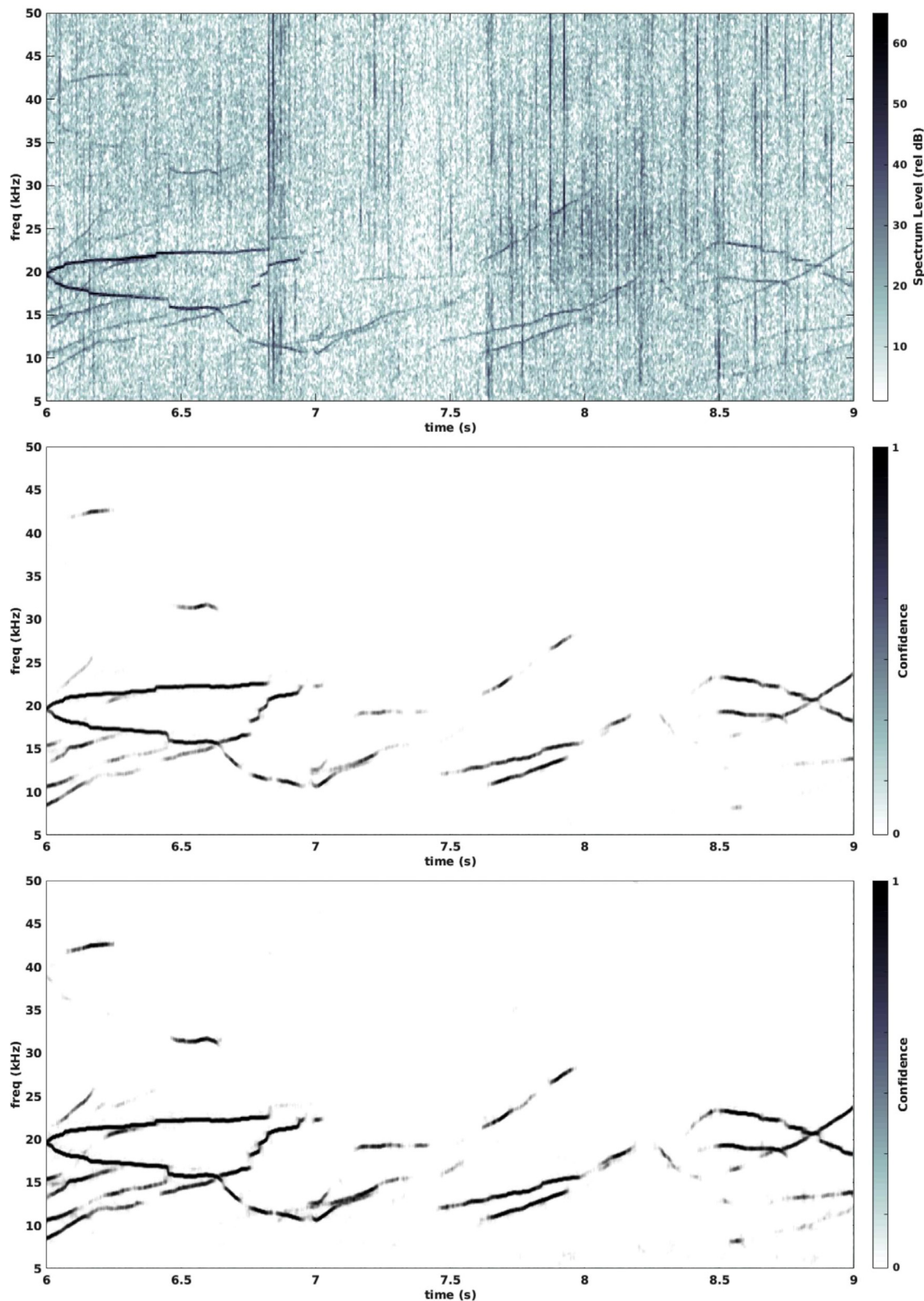


FIG. 7. (Color online) Comparison of CNN confidence map predictions from a spectrogram (top) using different loss functions. Predictions using L_{base} and L_{recall} loss are shown in the middle and lower panels, respectively.

whistle extraction recall of 85.93% and a precision of 89.50%. This is the target performance that we wish to achieve using pseudo-labels. By applying graph search on spectral peaks, graph search detects 72.28% (recall) of the analyst-annotated whistles with a precision of 81.13% ($F1$, 75.95%). As a competitive baseline, SMC-PHD detects 92.45% (recall) of annotated whistles with a precision of 76.55% ($F1$, 83.40%). After removing the detections that are shorter than 150 ms, SMC-PHD achieves a precision of 95.85% while the recall drops to 60.88% ($F1$, 74.38%), indicating that SMC-PHD is not retrieving longer whistles as well.

Replacing the analyst-annotation training data with pseudo-labels generated by graph search, " $L_{\text{base-graph}}$," extracts whistles with a recall of 61.53% and a precision of 94.15% ($F1$, 74.14%). When we trained the model with SMC-PHD detections and L_{base} , we obtained a recall of 72.75% and a precision of 94.98% ($F1$, 82.34%). Removing shorter detections resulted in a recall of 56.08% and a precision of 98.45% ($F1$, 70.97%). These models achieved

inferior $F1$ -scores than the corresponding methods for generating pseudo-labels.

In contrast, our modified loss function [Eq. (3)] leads to an $F1$ -score of 87.2% with $L_{\text{recall-SMCPHD} \geq 150\text{ms}}$, which is almost identical to the model trained with a large analyst-annotated dataset ($F1$ -score of 87.47%). Additionally, we observe relative improvements in coverage from 6.8 to 18.5% when we train models with L_{recall} as compared to models using the L_{base} loss function. We also have fewer fragments in detected whistles in our L_{recall} experiments. Combining these observations, our model trained with L_{recall} can correctly predict more t-f bins as whistles. Finally, although we observe a higher mean frequency deviation in pseudo-label experiments compared to graph search and SMC-PHD, the increase in deviation is less than one frequency bin width (125 Hz) on our spectrogram, which makes it negligible in subsequent applications that use detected tonals. Details of the performance on each species are summarized in Tables IV–VII in the Appendix.

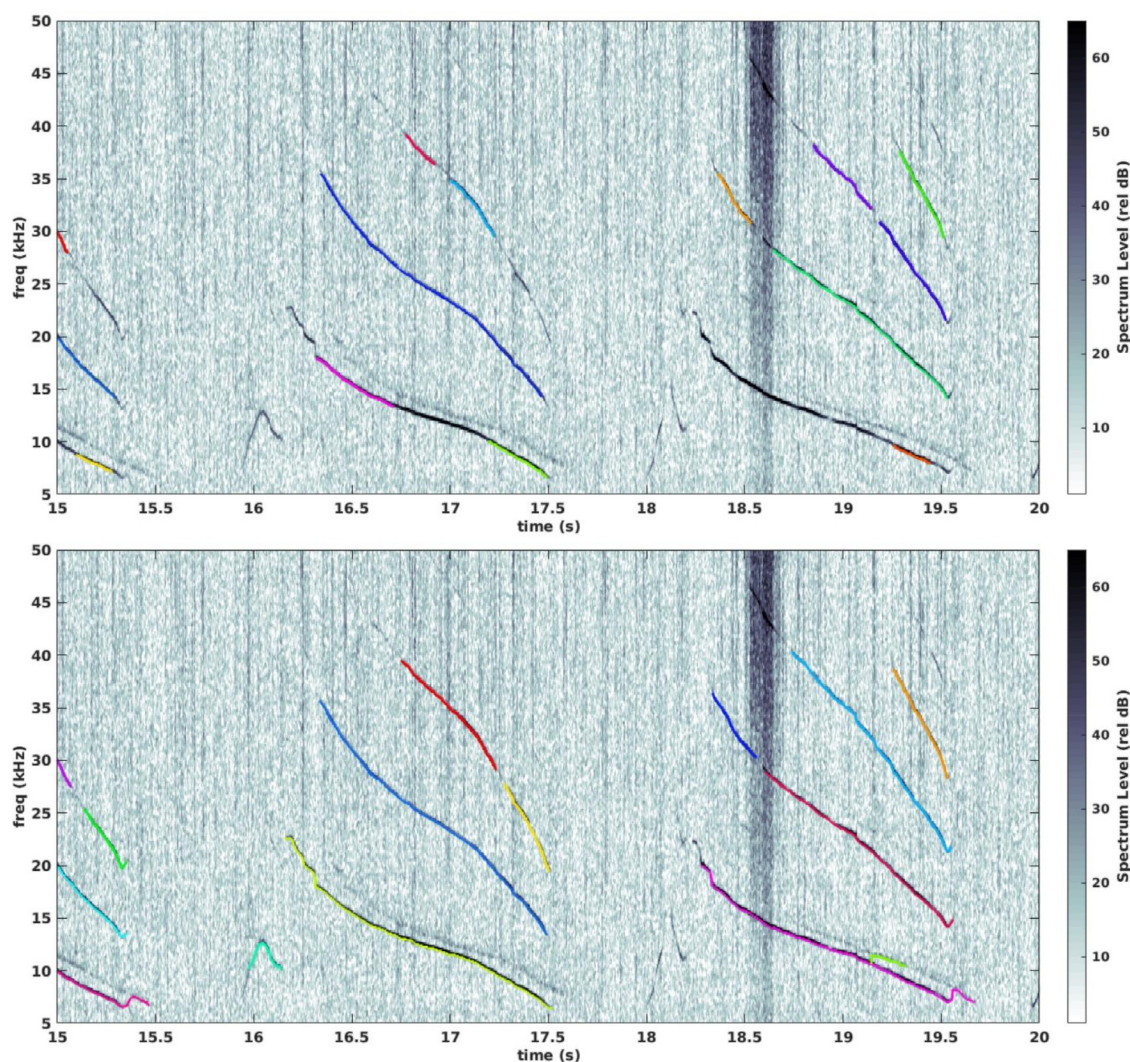


FIG. 8. (Color online) Comparison of detected bottlenose dolphin whistles among different experiments. Each whistle is colored differently. (Upper) Extracted whistles by CNN trained with L_{base} and (lower) extracted whistles by CNN trained with L_{recall} are shown.

B. Ablation study on pseudo-label generated by graph search

We design an ablation study for the loss function design (L_{base} , L_{recall} , L_{prec}) and hyperparameter choice as stated in Sec. III. Pseudo-labels were generated by graph search. We present the experiment results in Fig. 5, where each point reports the performance of one whistle extraction method. Points joined into a curve show the results for a fixed γ with variation of λ as specified in Sec. III (experiments section). The curves of L_{recall} show a tendency of increased recall and decreased precision when λ is increased. In contrast, larger λ in L_{prec} leads to higher precision but lower recall. The details of the performance are depicted in Tables VIII and IX in the Appendix.

For models trained with L_{recall} , we observe a significant increase in recall ($>21\%$ in the best case) compared to $L_{\text{base-graph}}$ while still achieving a reasonable precision (89.6%). For models trained with pseudo-labels and L_{prec} loss, we find that precision is slightly increased compared to the unaltered loss function, L_{base} , when we apply a larger value λ (0.01), but there is a slight decrease in recall. In comparison to graph search, precision is greatly increased at the cost of significant loss in recall. We observe that increasing λ leads to a decreasing $F1$ -score.

These comparisons show that the new loss metrics, L_{recall} and L_{prec} , can increase recall or precision with respect to the performance of the algorithm that generated the pseudo-labels. Models trained with L_{prec} lead to comparable $F1$ -scores with $L_{\text{base-graph}}$ while L_{recall} results in a significant $F1$ -score increase (12.17%). The $L_{\text{recall-graph}}$ produce results that are similar to the performance on human analyst-training data, and the $F1$ -score of 86.31% ($\lambda = 2$, $\gamma = 0$) approaches the $L_{\text{base-annotation}}$ $F1$ -score of 87.47%.

C. Ablation study on pseudo-labels generated by SMC-PHD

We perform a similar ablation study with models trained on SMC-PHD pseudo-labels. The precision-recall performance is shown in Fig. 6. We observe a similar trend on the curves of L_{recall} , where increased λ result in increased recall and decreased precision, and we achieve the best $F1$ -score using $\text{SMCPHD} \geq 150\text{ms}$ pseudo-label when $\lambda = 14$ and $\gamma = 1$. The maximum $F1$ -score is 87.2% in the experiment of “ $L_{\text{recall-SMCPHD} \geq 150\text{ms}}$,” which is a significant increase compared to “ $L_{\text{base-SMCPHD} \geq 150\text{ms}}$,” “ $L_{\text{base-SMCPHD}}$,” $\text{SMCPHD} \geq 150\text{ms}$, and SMCPHD . The details of the performance are noted in Tables X and XI in the Appendix.

D. Visualization of model output and whistle extraction result

We show examples of network output and extracted whistles produced by our algorithms in Figs. 7 and 8. We compare typical network outputs (confidence maps) of models trained with L_{recall} and L_{base} in Fig. 7. The model trained with L_{recall} had higher response to whistle energy and produced more continuous coverage of whistles compared to the model trained with L_{base} . When the confidence map predictions are

processed by *Silbido*’s graph search algorithm (and likely many other whistle extraction algorithms), this results in more and longer extracted whistles (Fig. 8). More examples of whistle extraction results are displayed in Fig. 10 in the Appendix.

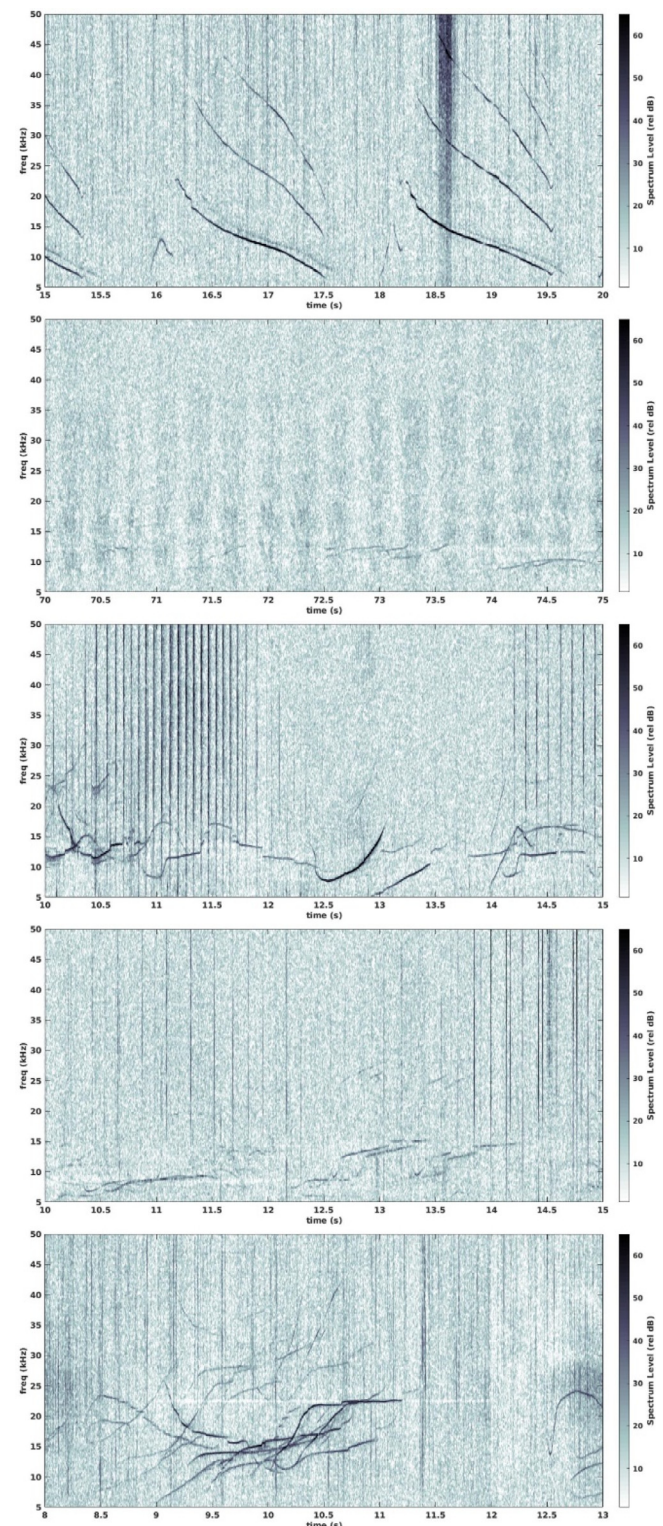


FIG. 9. (Color online) Examples of the whistle signals of five species, showing (top to bottom) Bottlenose dolphin, long-beaked common dolphin, short-beaked common dolphin, melon-headed whale, and spinner dolphin.

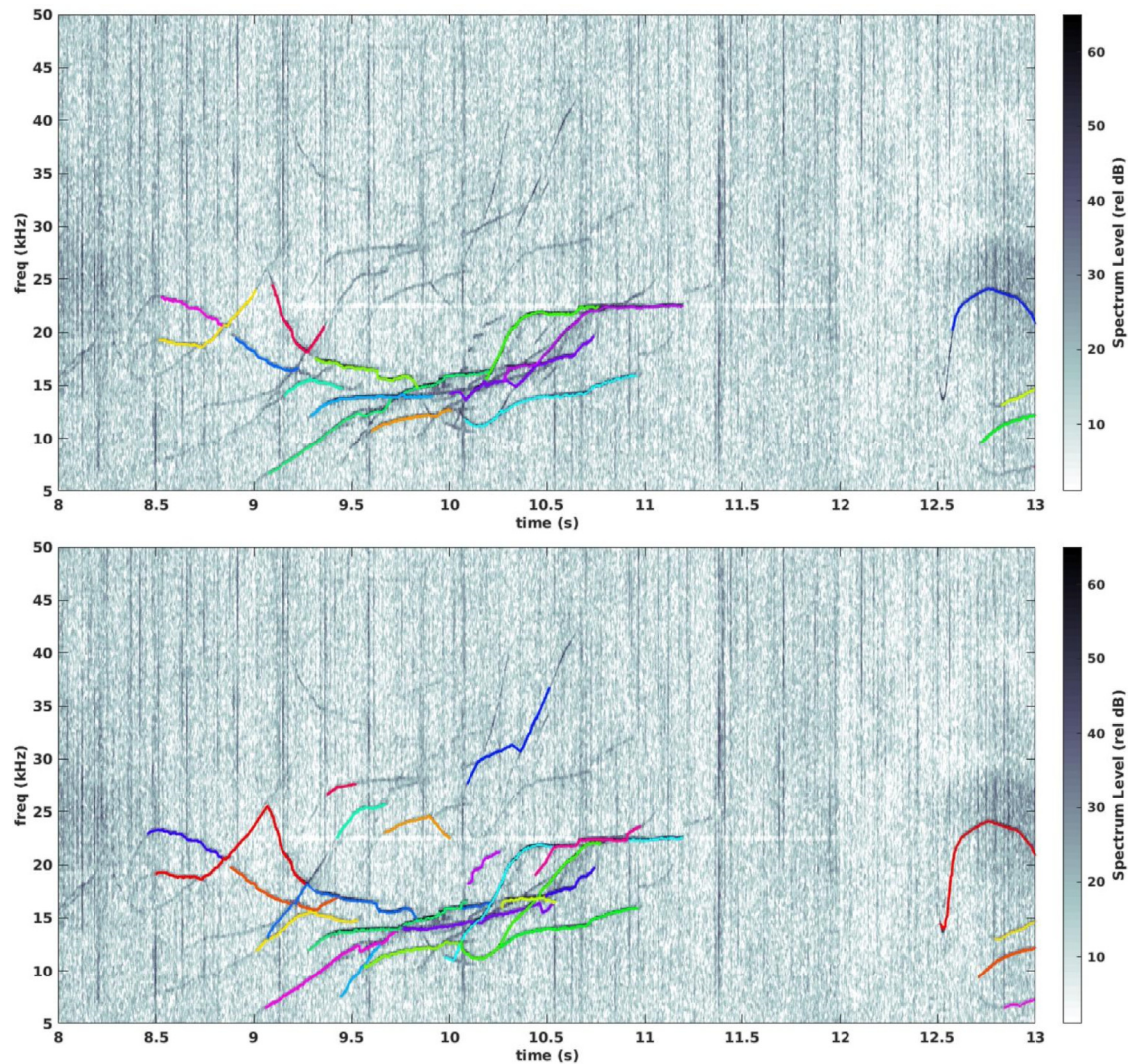


FIG. 10. (Color online) Comparison of detected spinner dolphin whistles among different experiments. Each whistle is colored differently. (Upper) Extracted whistles by CNN trained with L_{base} and (lower) extracted whistles by CNN trained with L_{recall} are shown.

V. DISCUSSION

Our results show that it is feasible to train competitive deep-learning-based models for whistle extraction without using analyst annotations as training data. With pseudo-labels generated by graph search and the proposed loss function, we can extract whistles with an $F1$ -score of 86.42%, which significantly surpasses graph search ($F1$ -score of 75.95%). With pseudo-labels generated by SMC-PHD, which uses 185 annotated whistles for training, we are able to further improve the whistle extraction performance to an $F1$ -score of 87.2%, which is close to the whistle extractor trained with 12 539 annotated whistles. By using different whistle extractors to generate pseudo-labels, the proposed method can eliminate or greatly reduce the human analyst annotation effort.

The comparison between the performance of the algorithms used to generate pseudo-labels and models trained on these labels with L_{base} loss shows that the models trained with the baseline loss function are fairly accurate in their

detections, but they miss many whistles that would have been detected by the algorithm that was used to generate the pseudo-labels. We also observe that SMC-PHD tends to extract more short whistle contours than graph search.

TABLE III. Summary of the specific DCLDE 2011 audio files used in the evaluation dataset.

Species	DCLDE 2011 Files
Bottlenose dolphin (<i>T. truncatus</i>)	Qx-Tt-SCI0608-N1-060814-121518 palmyra092007FS192-070924-205305 palmyra092007FS192-070924-205730
Long-beaked common dolphin (<i>D. capensis</i>)	Qx-Dc-CC0411-TAT11-CH2-041114-154040-s Qx-Dc-CC0411-TAT11-CH2-041114-154040-s QX-Dc-FLIP0610-VLA-061015-165000
Melon-headed whale (<i>P. electra</i>)	QX-Dc-FLIP0610-VLA-061015-165000 palmyra092007FS192-071004-032342 palmyra102006-061020-204327_4
Spinner dolphin (<i>S. longirostris</i>)	palmyra092007FS192-070927-224737 palmyra092007FS192-070927-224737 palmyra102006-061103-213127_4

TABLE IV. Summary of the performance on bottlenose dolphin data. Scores indicate the harmonic mean ($F1$) of precision and recall, the mean deviation in frequency from analyst annotations (μ_σ), the percentage of each whistle that was detected (coverage), and the mean number of connected segments for each whistle (fragmentation).

Method	$F1$ (%)	Precision (%)	Recall (%)	μ_σ (Hz)	Coverage (%)	Fragmentation (detected segments/whistle)
L_{base} —annotation	90.64	96.30	85.60	98.00	85.80	1.20
Graph search	88.06	92.30	84.20	112.00	79.30	1.20
SMCPHD	90.52	85.80	95.80	122.00	72.60	2.00
SMCPHD ≥ 150 ms	77.05	97.70	63.60	116.00	74.10	1.30
L_{base} —graph	69.17	96.20	54.00	170.00	72.30	1.40
L_{base} —SMCPHD	70.65	97.50	55.40	150.00	72.70	1.40
L_{base} —SMCPHD ≥ 150 ms	83.38	96.50	73.40	148.00	81.50	1.30
L_{recall} —graph	91.49	92.60	90.40	188.00	86.90	1.20
L_{recall} —SMCPHD	91.32	92.90	89.80	159.00	87.20	1.20
L_{recall} —SMCPHD ≥ 150 ms	91.90	94.10	89.80	159.00	86.80	1.20

The longer detections of SMC-PHD are more likely to be correct detections. Adjusting the time threshold for SMC-PHD might achieve a better $F1$ -score on our evaluation datasets, but this experiment is beyond the scope of this paper.

The L_{recall} loss showed strong $F1$ performance gains over the algorithms used to produce the pseudo-labels used to train the CNNs. We observe that the system improves whistle coverage and reduces fragmentation in quantitative (Table II) and qualitative results (Figs. 7 and 8). The longer whistle detections with fewer gaps may better facilitate downstream research, e.g., species identification. The L_{prec} loss, which was only tested on labels generated by an algorithm that tends to produce more false negatives than false positives, provided gains in precision at the cost of significant drops in recall on these data. We suspect that it would fare better on pseudo-label sets with higher false positive rates. In contrast, models trained using the baseline loss function, L_{base} , were unable to produce $F1$ -scores that exceeded the performance of the algorithms used to produce the pseudo-labels. When some labels are present (e.g., evaluation data), it is relatively simple to score the pseudo-labels and determine whether they are more likely to have

false positives or false negatives. When such labels are not available, manual inspection of pseudo-labels can provide intuition about which type of error is more prevalent.

There were likely more false negatives than false positives in our pseudo-labels for both label generation methods. We observed a higher precision than recall in graph search and SMCPHD ≥ 150 ms. Although SMCPHD has higher recall (92.45%) than precision (76.55%), the coverage was around 71%, suggesting that roughly 29% of t-f bins in whistles were not detected and the t-f bin-level recall was lower. Furthermore, because we balanced the number of negative patches and positive patches in the training dataset and false negative patches only covered a small portion of negative patches, many false negatives were excluded from training. While the proposed L_{recall} and L_{prec} were effective in improving whistle extraction recall or precision, respectively, L_{recall} increased $F1$ -score significantly more than L_{prec} . This observation also indicated that false negatives (missed whistles) in the pseudo-labels affect our whistle extraction model more than false positives.

The pseudo-label regularization terms demonstrated the ability to extract whistles that outperformed the algorithms

TABLE V. Summary of the performance on long-beaked common dolphin data. Scores indicate the harmonic mean ($F1$) of precision and recall, the mean deviation in frequency from analyst annotations (μ_σ), the percentage of each whistle that was detected (coverage), and the mean number of connected segments for each whistle (fragmentation).

Method	$F1$ (%)	Precision (%)	Recall (%)	μ_σ (Hz)	Coverage (%)	Fragmentation (detected segments/whistle)
L_{base} —annotation	83.98	89.90	78.80	96.00	87.50	1.10
Graph search	60.20	75.40	50.10	75.00	83.90	1.30
SMCPHD	73.60	62.90	88.70	103.00	69.80	1.70
SMCPHD ≥ 150 ms	70.27	93.20	56.40	96.00	73.80	1.20
L_{base} —graph	63.14	84.80	50.30	146.00	79.40	1.10
L_{base} —SMCPHD	60.65	99.60	43.60	105.00	73.20	1.10
L_{base} —SMCPHD ≥ 150 ms	72.30	86.90	61.90	113.00	81.80	1.10
L_{recall} —graph	73.81	77.20	70.70	151.00	86.50	1.20
L_{recall} —SMCPHD	74.44	70.70	78.60	122.00	86.20	1.10
L_{recall} —SMCPHD ≥ 150 ms	75.59	73.60	77.70	125.00	87.00	1.20

TABLE VI. Summary of the performance on melon-headed whale data. Scores indicate the harmonic mean ($F1$) of precision and recall, the mean deviation in frequency from analyst annotations (μ_σ), the percentage of each whistle that was detected (coverage), and the mean number of connected segments for each whistle (fragmentation).

Method	$F1$ (%)	Precision (%)	Recall (%)	μ_σ (Hz)	Coverage (%)	Fragmentation (detected segments/whistle)
$L_{\text{base}}-\text{annotation}$	82.11	77.50	87.30	86.00	91.70	1.10
Graph search	69.20	66.70	71.90	95.00	80.50	1.10
SMCPHD	77.56	67.20	91.70	92.00	69.10	1.50
$\text{SMCPHD} \geq 150 \text{ ms}$	70.49	95.40	55.90	88.00	74.10	1.10
$L_{\text{base}}-\text{graph}$	78.75	97.60	66.00	135.00	80.00	1.10
$L_{\text{base}}-\text{SMCPHD}$	70.83	98.50	55.30	109.00	75.00	1.10
$L_{\text{base}}-\text{SMCPHD} \geq 150 \text{ ms}$	85.26	98.60	75.10	113.00	83.80	1.10
$L_{\text{recall}}-\text{graph}$	88.74	93.30	84.60	142.00	88.80	1.10
$L_{\text{recall}}-\text{SMCPHD}$	87.41	89.40	85.50	122.00	89.70	1.10
$L_{\text{recall}}-\text{SMCPHD} \geq 150 \text{ ms}$	89.28	92.70	86.10	123.00	89.40	1.10

TABLE VII. Summary of the performance on spinner dolphin data. Scores indicate the harmonic mean ($F1$) of precision and recall, the mean deviation in frequency from analyst annotations (μ_σ), the percentage of each whistle that was detected (coverage), and the mean number of connected segments for each whistle (fragmentation).

Method	$F1$ (%)	Precision (%)	Recall (%)	μ_σ (Hz)	Coverage (%)	Fragmentation (detected segments/whistle)
$L_{\text{base}}-\text{annotation}$	93.14	94.30	92.00	88.00	87.30	1.10
Graph search	86.35	90.10	82.90	122.00	80.50	1.30
SMCPHD	91.92	90.30	93.60	115.00	72.20	2.00
$\text{SMCPHD} \geq 150 \text{ ms}$	79.71	97.10	67.60	114.00	69.50	1.30
$L_{\text{base}}-\text{graph}$	85.48	98.00	75.80	128.00	78.30	1.10
$L_{\text{base}}-\text{SMCPHD}$	81.74	98.20	70.00	115.00	73.90	1.20
$L_{\text{base}}-\text{SMCPHD} \geq 150 \text{ ms}$	88.41	97.90	80.60	117.00	79.90	1.20
$L_{\text{recall}}-\text{graph}$	91.20	95.10	87.60	136.00	84.70	1.20
$L_{\text{recall}}-\text{SMCPHD}$	92.50	95.70	89.50	134.00	86.10	1.20
$L_{\text{recall}}-\text{SMCPHD} \geq 150 \text{ ms}$	92.03	94.70	89.50	132.00	86.00	1.2

TABLE VIII. Summary of the performance for altering γ and λ in $L_{\text{recall}}-\text{graph}$.

γ	λ	$F1$ (%)	Precision (%)	Recall (%)
0	0.5	79.30	90.18	70.93
	1	83.14	88.68	78.38
	2	86.31	89.55	83.33
	3	84.38	84.08	84.78
	4	82.01	79.33	85.18
1	2	83.37	88.70	78.85
	4	84.80	86.90	82.95
	6	84.88	86.08	83.85
	8	81.61	78.68	85.20
2	4	83.52	87.85	79.85
	6	83.88	85.75	82.48
	8	84.39	85.78	83.30
4	10	83.98	83.30	84.90
	4	81.87	91.20	74.35
	6	82.47	89.95	76.28
	8	82.82	88.88	77.80
	10	83.03	88.28	78.70
	15	83.30	85.35	81.98
	20	83.59	83.63	84.43
	25	81.08	79.90	83.88
	30	79.83	76.83	85.10

TABLE IX. Summary of the performance for altering γ and λ in $L_{\text{prec}}-\text{graph}$.

γ	λ	$F1$ (%)	Precision (%)	Recall (%)
0	0.01	74.51	93.88	62.20
	0.1	72.25	95.73	58.53
1	0.01	74.52	93.53	62.33
	0.1	72.39	96.03	58.80
2	0.01	74.55	94.80	61.83
	0.1	72.16	95.55	58.58
4	0.01	74.81	95.38	61.95
	0.1	72.85	95.68	59.40

TABLE X. Summary of the performance for altering γ and λ in $L_{\text{recall}}-\text{SMCPHD}$.

γ	λ	$F1$ (%)	Precision (%)	Recall (%)
0	1	85.65	89.35	82.40
	2	86.42	87.18	85.85
	3	86.22	85.15	87.34
1	4	85.81	83.63	88.20
	2	85.16	89.88	81.10
	4	86.14	88.80	84.05
	6	86.26	88.28	84.60
	8	86.2	87.18	85.40
	10	85.9	85.73	86.35
	11	85.94	84.90	87.10
	12	86.09	84.40	87.95

TABLE XI. Summary of the performance for altering γ and λ in $L_{\text{recall-SMCPHD} \geq 150\text{ms}}$.

γ	λ	$F1$ (%)	Precision (%)	Recall (%)
0	1	81.95	93.88	72.78
	2	84.91	91.98	78.88
	3	85.92	89.43	82.95
	4	86.82	89.23	84.65
	5	86.16	87.40	85.10
	6	86.48	86.33	86.78
1	2	81.58	95.40	71.35
	4	84.03	92.10	77.30
	6	85	90.95	79.93
	8	86.11	90.68	82.13
	10	86.74	90.28	83.68
	11	86.51	89.45	83.98
	12	86.77	88.40	85.48
	14	87.2	88.78	85.78
	16	86.38	87.33	85.58

used to produce the pseudo-labels. Due to whistle t-f nodes labeled as noise in the pseudo-labels (e.g., Fig. 3), learning without regularization produced models that had high precision but sacrificed the ability to produce high confidence map scores for many whistles, resulting in low recall. The regularization in $L_{\text{recall-graph}}$ sacrifices some of the precision attained with $L_{\text{base-graph}}$ but compensates with a much higher recall, leading to an overall superior $F1$ -score (Table II). Similar trends are observed with pseudo-labels generated by SMC-PHD.

VI. CONCLUSION

We have developed a convolutional DNN that can be trained without any analyst annotations and is able to extract whistles with a performance comparable to one trained from a rich set of analyst annotations. Instead of using the expensive and time-consuming annotations produced by analysts, we used methods that required no (graph search) or minimal training data (SMCPHD) to extract whistle annotations used as pseudo-labels for model training. We evaluated extraction performance on a diverse four-species evaluation dataset consisting of 1935 analyst-annotated whistles (duration $\geq 150\text{ms}$; $\geq 1/3$ of the t-f bins have a $\text{SNR} \geq 10\text{dB}$). Performance of a baseline CNN model using a standard loss function (L_{base}) produced $F1$ -scores comparable to the performance of the algorithms used to produce the pseudo-labels. However, there was a tendency to increase precision at a nontrivial cost to recall.

The proposed loss functions significantly improve whistle extraction performance. Regularization penalties compensated for errors in pseudo-labels and prioritized recall [L_{recall} , Eq. (3)] or precision [L_{prec} , Eq. (5)]. Our experiments demonstrated that missed whistles in pseudo-labels affect the CNN model more than the incorrectly detected whistles, and the proposed L_{recall} loss function outperformed L_{base} with an absolute $F1$ -score increase of 12.17% (graph search pseudo-labels) and 3.8% (SMC-PHD pseudo-labels). In the

best case, a model trained without any analyst annotations using SMC-PHD detections of at least 150ms duration detected 85.78% of the whistles with a precision of 88.78%. The $F1$ -score (87.2%) was comparable to a model trained with 12 539 annotated whistles (87.47%), showing the potential to generate whistle extraction models with near state-of-the-art performance with little to no human annotation effort.

ACKNOWLEDGMENTS

We thank the DCLDE organizers for providing the DCLDE 2011 dataset used in this work, as well as the numerous crews and science staff responsible for hardware development and deployment, visual observations, and annotation that resulted in these public datasets. Our thanks to Dr. Michael Weise, Office of Naval Research, for financial support (Grant No. N00014-21-1-2567). We also thank the anonymous reviewers who contributed insightful suggestions that improved this manuscript.

APPENDIX

See Figs. 9 and 10 as well as Tables III–XI for additional details of the experiments conducted in this study.

¹We used beta2 version of *Silbido* at <https://roch.sdsu.edu/index.php/software/>. The latest version of *Silbido* is available at <https://github.com/MarineBioAcousticsRC/silbido> (Last viewed July 18, 2023).

²The preprocessing code is available at <https://doi.org/10.5258/SOTON/D0316>. The SMC-PHD code is available at https://github.com/PinaGruden/SMC-PHD_whistle_contour_tracking (Last viewed July 18, 2023).

³Code is available at <https://github.com/Paul-LiPu/DeepWhistle> (Last viewed July 18, 2023).

- Bradski, G. (2000). "The OpenCV Library," Dr. Dobb's J. Software Tools **25**(11), 122–125.
- Caldwell, M. C., and Caldwell, D. K. (1968). "Vocalization of naive captive dolphins in small groups," *Science* **159**, 1121–1123.
- Conant, P. C., Li, P., Liu, X., Klinck, H., Fleishman, E., Gillespie, D., Nosal, E.-M., and Roch, M. A. (2022). "Silbido profundo: An open source package for the use of deep learning to detect odontocete whistles," *J. Acoust. Soc. Am.* **152**, 3800–3808.
- DCLDE Organizing Committee (2011). "Detection, classification, localization, and density estimation (DCLDE) of marine mammals using passive acoustic monitoring workshop dataset," available at https://www.moby-sound.org/workshops_p2.html (Last viewed 2023-02-13).
- Dillon, W. R., and Goldstein, M. (1984). *Multivariate Analysis, Methods and Applications* (Wiley, New York).
- Fisher, F. H., and Spiess, F. N. (1963). "FLIP-floating instrument platform," *J. Acoust. Soc. Am.* **35**, 1633–1644.
- Ghosh, A., Kumar, H., and Sastry, P. S. (2017). "Robust loss functions under label noise for deep neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, February 4–9, San Francisco, CA (AAAI Press, Palo Alto, CA), Vol. 31.
- Gillespie, D., Caillat, M., Gordon, J., and White, P. (2013). "Automatic detection and classification of odontocete whistles," *J. Acoust. Soc. Am.* **134**, 2427–2437.
- Goldberger, J., and Ben-Reuven, E. (2017). "Training deep neural-networks using a noise adaptation layer," in *International Conference on Learning Representations*, April 24–26, Toulon, France (Curran Associates, Inc., Red Hook, NY), p. 9.

- Gruden, P., and White, P. R. (2016). "Automated tracking of dolphin whistles using Gaussian mixture probability hypothesis density filters," *J. Acoust. Soc. Am.* **140**, 1981–1991.
- Gruden, P., and White, P. R. (2020). "Automated extraction of dolphin whistles—A sequential Monte Carlo probability hypothesis density approach," *J. Acoust. Soc. Am.* **148**, 3014–3026.
- He, K., Zhang, X., Ren, S., and Sun, J. (2015). "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, December 7–13, Santiago, Chile (IEEE, New York), pp. 1026–1034.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 27–30, Las Vegas, NV (IEEE, New York), pp. 770–778.
- Ioffe, S., and Szegedy, C. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, PMLR (37), July 6–11, Lille, France (Proceedings of Machine Learning Research, Albany, NY), pp. 448–456.
- Janik, V. M., King, S. L., Sayigh, L. S., and Wells, R. S. (2013). "Identifying signature whistles from recordings of groups of unrestrained bottlenose dolphins (*Tursiops truncatus*)," *Mar. Mammal Sci.* **29**, 109–122.
- Jiang, J.-J., Bu, L.-R., Duan, F.-J., Wang, X.-Q., Liu, W., Sun, Z.-B., and Li, C.-Y. (2019). "Whistle detection and classification for whales based on convolutional neural networks," *Appl. Acoust.* **150**, 169–178.
- Kaplan, M. B., Aran Mooney, T., Sayigh, L. S., and Baird, R. W. (2014). "Repeated call types in Hawaiian melon-headed whales (*Peponocephala electra*)," *J. Acoust. Soc. Am.* **136**, 1394–1401.
- Kim, Y., Yim, J., Yun, J., and Kim, J. (2019). "NLNL: Negative learning for noisy labels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 27–November 2, Seoul, South Korea (IEEE, New York), pp. 101–110.
- Kim, Y., Yun, J., Shon, H., and Kim, J. (2021). "Joint negative and positive learning for noisy labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 20–25, Nashville, TN (IEEE, New York), pp. 9442–9451.
- Li, P., Liu, X., Palmer, K., Fleishman, E., Gillespie, D., Nosal, E.-M., Shiu, Y., Klinck, H., Cholewiak, D., and Helble, T. (2020). "Learning deep models from synthetic data for extracting dolphin whistle contours," in *2020 International Joint Conference on Neural Networks (IJCNN)*, July 19–24, Glasgow, UK (IEEE, New York), pp. 1–10.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, October 22–29, Venice, Italy (IEEE, New York), pp. 2980–2988.
- Ma, X., Huang, H., Wang, Y., Romano, S., Erfani, S., and Bailey, J. (2020). "Normalized loss functions for deep learning with noisy labels," in *Proceedings of Machine Learning Research, International Conference on Machine Learning*, July 13–18, virtual, PMLR (119) pp. 6543–6553.
- Mallawaarachchi, A., Ong, S., Chitre, M., and Taylor, E. (2008). "Spectrogram denoising and automated extraction of the fundamental frequency variation of dolphin whistles," *J. Acoust. Soc. Am.* **124**, 1159–1170.
- Mellinger, D. K., Martin, S. W., Morrissey, R. P., Thomas, L., and Yosco, J. J. (2011). "A method for detecting whistles, moans, and other frequency contour sounds," *J. Acoust. Soc. Am.* **129**, 4055–4061.
- Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013). "Learning with noisy labels," in *Advances in Neural Information Processing Systems*, December 5–8, Lake Tahoe, NV (Curran Associates Inc., Red Hook, NY), Vol. 26.
- Oswald, J. N., Barlow, J., and Norris, T. F. (2003). "Acoustic identification of nine delphinid species in the eastern tropical Pacific Ocean," *Mar. Mammal Sci.* **19**, 20–37.
- Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. (2017). "Making deep neural networks robust to label noise: A loss correction approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, July 21–26, Honolulu, HI (IEEE, New York), pp. 1944–1952.
- Roch, M. A., Scott Brandes, T., Patel, B., Barkley, Y., Baumann-Pickering, S., and Soldevilla, M. S. (2011). "Automated extraction of odontocete whistle contours," *J. Acoust. Soc. Am.* **130**, 2212–2223.
- Sayigh, L., Quick, N., Hastie, G., and Tyack, P. (2013). "Repeated call types in short-finned pilot whales, *Globicephala macrorhynchus*," *Mar. Mammal Sci.* **29**, 312–324.
- Sjare, B. L., and Smith, T. G. (1986). "The relationship between behavioral activity and underwater vocalizations of the white whale, *Delphinapterus leucas*," *Can. J. Zool.* **64**, 2824–2831.
- Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. (2022). "Learning from noisy labels with deep neural networks: A survey," *IEEE Trans. Neural Networks Learn. Syst.* (published online).
- Su, Z., Liu, W., Yu, Z., Hu, D., Liao, Q., Tian, Q., Pietikäinen, M., and Liu, L. (2021). "Pixel difference networks for efficient edge detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 10–17 (IEEE, New York), pp. 5117–5127.
- Tanno, R., Saeedi, A., Sankaranarayanan, S., Alexander, D. C., and Silberman, N. (2019). "Learning from noisy labels by regularized estimation of annotator confusion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 15–20, Long Beach, CA (IEEE, New York), pp. 11244–11253.
- Taruski, A. G. (1979). "The whistle repertoire of the North Atlantic pilot whale (*Globicephala melaena*) and its relationship to behavior and environment," in *Behavior of Marine Animals* (Springer, New York), pp. 345–368.
- van Parijs, S. M., and Corkeron, P. J. (2001). "Evidence for signature whistle production by a Pacific humpback dolphin, *Sousa chinensis*," *Mar. Mammal Sci.* **17**, 944–949.
- Wang, X., Hua, Y., Koldirov, E., and Robertson, N. M. (2019a). "IMAE for noise-robust learning: Mean absolute error does not treat examples equally and gradient magnitude's variance matters," *arXiv:1903.12141*.
- Wang, X., Jiang, J., Duan, F., Liang, C., Li, C., Sun, Z., Lu, R., Li, F., Xu, J., and Fu, X. (2021). "A method for enhancement and automated extraction and tracing of Odontoceti whistle signals base on time-frequency spectrogram," *Appl. Acoust.* **176**, 107698.
- Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. (2019b). "Symmetric cross entropy for robust learning with noisy labels," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, October 27–November 2, Seoul, South Korea (IEEE, New York), pp. 322–330.
- White, P., and Hadley, M. (2008). "Introduction to particle filters for tracking applications in the passive acoustic monitoring of cetaceans," *Can. Acoust.* **36**, 146–152.
- Wursig, B., and Perrin, W. F. (2009). *Encyclopedia of Marine Mammals* (Academic, New York).
- Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., and Sugiyama, M. (2019). "Are anchor points really indispensable in label-noise learning?," in *Advances in Neural Information Processing Systems*, December 8–14, Vancouver, Canada (Curran Associates Inc., Red Hook, NY), pp. 6835–6846.
- Zhang, Z., and Sabuncu, M. (2018). "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in Neural Information Processing Systems*, February 2–7, New Orleans, LA (Curran Associates Inc., Red Hook, NY), Vol. 31.