JASA ARTICLE

Acoustic target recognition of large yellow croaker via small-sample deep learning with a proposed dataset partitioning strategy

Yi Tao,[1,2,3] Hanxi Jiang,[1,2,3] Xuchen Wang,[1,2,3] Yihao Zhao,[1,2,3] Chao Li,[4] and Yougan Chen[1,2,3,5,a)]

[1]*Key Laboratory of Underwater Acoustic Communication and Marine Information Technology (Xiamen University), Ministry of Education, Xiamen 361005, China*

[2]*Shenzhen Research Institute of Xiamen University, Shenzhen 518000, China*

[3]*College of Ocean and Earth Sciences, Xiamen University, Xiamen 361102, China*

[4]*State Key Laboratory of Acoustics and Marine Information, Chinese Academy of Sciences, Beijing 100190, China*

[5]*Fujian Ocean Innovation Center, Xiamen 361102, China*

ABSTRACT:

Research on acoustic target recognition of the large yellow croaker (*Larimichthys crocea*) holds significant implications for precise monitoring of their population status and maintaining marine ecological balance. Due to the complex acoustic fields in underwater aquaculture environments and the high time cost of data collection, it is often challenging to gather sufficient effective samples for target recognition tasks. To address the practical challenges in small-sample scenarios—such as limited effective acoustic data, high recognition difficulty, low accuracy, and sensitivity to anomalous samples—this study systematically integrates and adapts a set of established techniques, including data partitioning and ensemble learning, specifically tailored for the acoustic characteristics of the large yellow croaker in aquaculture environments. The proposed approach employs a pre-grouping strategy on the training dataset and incorporates a loss-based weighting mechanism to adjust sub-model contributions, with further optimization focused on efficient data partitioning under small-sample conditions. Experimental results on a dedicated small-sample acoustic dataset of the large yellow croaker demonstrate that the method achieves a recognition F1-score of 87.8% and helps mitigate feature-learning imbalances, indicating its practical effectiveness for this specific application. https://doi.org/10.1121/10.0043129

(Received 7 June 2025; revised 15 February 2026; accepted 24 February 2026; published online 14 April 2026)

[Editor: Zoi-Heleni Michalopoulou] Pages: 3291–3310

I. INTRODUCTION

The large yellow croaker holds paramount importance as a marine fishery resource in China, sustaining coastal economies while serving as a keystone species in subtropical marine ecosystems.[1] Its distinct vocalization patterns, generated through coordinated contractions of specialized sonic muscles and swim bladder vibrations,[2] provide critical biomarkers for population monitoring. Spectral analyses reveal species-specific acoustic signatures with dominant energy near 800 Hz, exhibiting temporal waveform variations correlated with feeding (single pulses, 1–30 ms intervals) and spawning behaviors (multi-pulse sequences, 100–130 ms intervals).[3] These bioacoustics features underpin underwater acoustic target recognition (UATR) systems that enable non-invasive assessment of distribution dynamics and abundance trends,[4] forming the scientific basis for sustainable fishery management.

Conventional UATR methodologies predominantly rely on data-intensive deep learning models, requiring substantial training samples to achieve robust recognition accuracy.[5,6] However, underwater acoustic data acquisition faces inherent physical constraints: compared to atmosphere, water exhibits 800× higher density, and the complex acoustic propagation environment causes severe acoustic attenuation and distortion due to absorption by the aqueous medium, scattering from suspended particles, and boundary reflections.[7] This energy dissipation, exacerbated by suspended particulates and boundary reflections,[8] limits effective propagation distances and reduces signal-to-noise ratios (SNRs) below detection thresholds in aquaculture environments.[9] In individual identification tasks, the limited life cycle of individuals is one of the constraints on sample accumulation. Consequently, obtaining sufficient training data demands prolonged monitoring cycles—a critical barrier for timely management decisions. For the large yellow croaker aquaculture industry, timely assessments during the growth cycle contribute to disease prevention and feeding optimization, potentially avoiding economic losses caused by disease outbreaks in critical situations.[10]

Therefore, shortening the data acquisition cycle and rapidly assessing growth conditions are imperative, as they facilitate adjustments to feeding regimes, water quality management, and other husbandry practices. A compressed data acquisition cycle

a)Email: chenyougan@xmu.edu.cn

provides more timely insights, better aligning with the demands of aquaculture management. Simultaneously, it enables earlier detection of disease symptoms, allowing prompt preventive measures to mitigate losses[11]—a crucial advantage in Small Sample Learning (SSL) contexts where limited training data exacerbates the urgency for efficient and actionable information. SSL refers to strategies in machine learning where insufficient training samples hinder effective model development, with the goal of constructing generalized models capable of solving target problems using minimal data.[12] The traditional processing methods of SSL mainly focus on removing various random noises in the atmosphere. The underwater acoustic channel is a typical frequency-selective fading channel, where the attenuation coefficient is strongly dependent on frequency, approximately proportional to the square of the frequency.[13] Within the 200 to 1200 Hz band, which is the primary focus of this study, the effects cannot be generalized. In particular, across the critical 500–800 Hz sub-band, signals in typical shallow-water aquaculture environments experience not only unavoidably high absorption loss but are also highly susceptible to multipath effects and ambient noise. This leads to significant signal distortion and a reduction in SNR.[3] In addition, underwater SSL must also confront the problem of ocean reverberation. The traditional SSL processing methods do not take into account the influence of these underwater sound fields.

Based on the above challenges, this paper presents a tailored deep learning approach for small-sample acoustic target recognition of large yellow croakers, which systematically incorporates a data-grouping strategy. We employ a fine-grained classification paradigm that treats individual identities as separate categories to achieve individual identification. Prior to the global training phase, the training dataset is partitioned into subgroups for independent pre-training. These sub-models are first validated to ensure baseline recognition capabilities before integration, thereby mitigating the adverse effects of limited sample size. The primary contributions of this work are as follows:

(1) A grouped training strategy is implemented to enhance data utilization in small-sample scenarios specific to large yellow croaker acoustic recognition. This interleaved grouping approach aims to alleviate the performance degradation commonly caused by scarce training data.
(2) A subgroup-specific training mechanism is designed to isolate potential anomalous samples, preventing their negative influence from propagating across different data groups during optimization.
(3) A loss-aware weighting scheme is integrated into the backpropagation process. This allows the model to dynamically adjust the influence of each subgroup during gradient updates, effectively reducing the impact of groups containing anomalous or low-quality samples.
(4) The influence of group quantity on model performance is empirically analyzed. Practical guidelines for data allocation in small-sample settings are provided, along with dataset-adaptive model refinements. These combined optimizations contribute to improved recognition accuracy under data constraints.

The remainder of this paper is organized as follows: Section I reviews related work in UATR. Section II presents the system model and relevant parameters. Section III details the implementation of the proposed group-based small-sample deep learning method, including algorithmic steps and model configuration discussions. Section IV describes the data sources and experimental results. Finally, Section V concludes the paper.

II. RELATED WORK

Currently, numerous machine learning methods have been widely applied in the field of hydroacoustic recognition, primarily focusing on sonar image recognition and time-frequency feature analysis. However, the application and broader adoption of SSL in UATR remain in their nascent stages, with most approaches leveraging self-supervised learning and transfer learning to address dataset inadequacies.

In the domain of sonar image recognition, Refs. 14–16 collectively advance technical approaches. Reference 14 systematically evaluated three SSL frameworks (Rotation Network, denoising autoencoders, and Jigsaw puzzles) on unlabeled real-world sonar datasets through pretraining and transfer learning. The Jigsaw method achieved 97.02% accuracy with 200 samples per class, only 1.35% below supervised baselines, demonstrating SSL's efficacy in small-sample scenarios. Reference 15 compared metric-based methods (Siamese/triplet networks) and library-based strategies across custom and public SeabedObjectsKLSG datasets, highlighting SSL's potential under data scarcity. Reference 16 innovatively applied Siamese networks to UATR, quantifying inter-class disparities using 80 single-beam trajectories across eight object types. The study validated stable target tracking through multi-beam mapping discrepancies, offering critical insights for deploying deep learning in UATR applications.

In the field of hydroacoustic time-frequency feature recognition, several studies addressing small-sample challenges have also emerged. Reference 17 integrated large pre-trained neural networks with customized acoustic attention modules to tackle small-sample datasets. The Scale ResNet module accepts Constant-Q transform (CQT) features as input to prioritize frequency-specific information, whereas the RHAF (Residual Hybrid Attention Fusion) module combines temporal features extracted by wav2vec 2.0 with frequency features from Scale ResNet, leveraging attention mechanisms to fuse time-frequency and temporal features synergistically. This adaptation enables speech-trained wav2vec 2.0 models to better generalize to hydroacoustic data. Experiments on the ShipsEar dataset demonstrated a recognition accuracy of 96.39%. Reference 18 developed Transfer-VGG16 (Visual Geometry Group 16), a novel transfer learning approach based on the VGG16 model utilizing three-dimensional data inputs. Evaluations on

https://doi.org/10.1121/10.0043129

real-world datasets confirmed its superior performance under scarce observational data. Reference 19 proposed a self-supervised dual-channel self-attention acoustic encoder (DSAE) for UATR tasks. This method unifies features into self-supervised learning via the dual-channel encoder, augmented by a dynamic positive sample memory module (DMM) to ensure comprehensive and balanced training sample utilization. Experimental results indicated that DSAE significantly outperforms state-of-the-art acoustic learning methods in recognition accuracy.

These studies collectively demonstrate that advanced machine learning techniques enable high-performance target recognition in the field of small-sample hydroacoustic target recognition, even under data scarcity. However, these works primarily achieved their results through transfer learning and self-supervised learning, which are prone to model overfitting.[20] Overfitted models tend to fixate on specific samples or noise in the training data rather than learning the true data distribution, thereby struggling to adapt to data variations. Moreover, existing literature has not adequately addressed the heightened sensitivity of underwater deep learning models to noise and outliers in small-sample scenarios, which may severely degrade model performance.[21]

Thus, although machine learning methods have proven effective in hydroacoustic target recognition,[14–19] the application of SSL in this domain remains fraught with challenges. In this paper, we propose a new small-sample deep learning method for UATR based on data grouping, named the Small-Sample Unbalanced Dataset Grouping method (SUDG), in audio signal-driven scenarios, incorporating analyses of model quantity configuration and dataset allocation strategies under small-sample constraints. Experimental validations in croaker environments further substantiate its efficacy.

III. UNDERWATER ACOUSTIC PROPAGATION SCENE AND SMALL SAMPLE MODEL

A. Underwater acoustic interference

The complexity of marine environments results in hydroacoustic fields characterized by diverse and variable interference types with varying intensities.[22] A single type of feature extraction cannot resolve all interference challenges. Current UATR tasks predominantly employ Mel-Frequency Cepstral Coefficients (MFCC) feature extraction, which processes time-domain signals according to the auditory sensitivity curve.[23] This relationship can be mathematically represented by the following formula:[24]

$$Mel(f) = 2595 * log_{10}\left(1 + \frac{f}{700}\right), \tag{1}$$

where f denotes the actual frequency, and $Mel(f)$ represents the perceptual frequency in Mels. The features derived from this method exhibit enhanced robustness.[25]

Although the MFCC method has been validated for its robustness and noise resistance in the field of hydroacoustic recognition,[26] strong interference scenarios still significantly impact the outcomes of MFCC feature extraction. This section elucidates the specific effects of interference on the MFCC feature extraction process through formalized derivations based on its underlying mechanism.

1. Ocean noise

Gaussian white noise, a prevalent form of additive interference, is typically expressed as Eq. (1),[27] where $y(t)$ denotes the received signal, $x(t)$ represents the original signal, and $n(t)$ corresponds to the Gaussian white noise component

$$y(t) = x(t) + n(t). \tag{2}$$

In the frequency domain, the power spectrum of the received signal is expressed as

$$|Y(f,t)|^2 = |X(f,t)|^2 + |N(f,t)|^2 + 2 \cdot Re\{X(f,t) \cdot N^*(f,t)\}. \tag{3}$$

Due to the stochastic nature of noise, the expectation of the cross term is zero; consequently, the expectation of the power spectrum is expressed as

$$E[|Y(f,t)|^2] = |X(f,t)|^2 + \sigma^2, \tag{4}$$

where σ^2 denotes the variance of the noise. After processing through the Mel filter bank, the Mel-spectral energy is expressed as

$$E_m = \sum_f |Y(f,t)|^2 \cdot Mel(f). \tag{5}$$

The introduction of noise increases the Mel-spectral energy, particularly under low SNR conditions, causing deviations in MFCC coefficients from their original values.[28]

2. Impact of multipath interference

Multipath interference, a convolutional disturbance caused by signal propagation through multiple paths, results in a received signal $y(t)$ expressed as

$$y(t) = \sum_{i=0}^{L-1} a_i \cdot x(t - \tau_i), \tag{6}$$

where a_i and τ_i denote the attenuation coefficient and time delay of the i-th path, respectively. In the frequency domain, the power spectrum of the received signal is expressed as

$$|Y(f,t)|^2 = \sum_{i=0}^{L-1} |a_i|^2 \cdot |X(f,t)|^2 + \sum_{i \neq j} a_i a_j^* \cdot X(f,t)X^*(f,t) \cdot e^{-j2\pi f(\tau_i - \tau_j)}. \tag{7}$$

Multipath interference introduces additional fluctuations in the frequency domain, altering the Mel-spectral energy distribution and ultimately degrading the accuracy of MFCC coefficients.

3. Impact of reverberation

Reverberation, resulting from the reflections and superposition of acoustic waves in enclosed spaces, can be modeled as the convolution of the original signal $x(t)$ with the room impulse response $h(t)$:

$$y(t) = x(t) * h(t). \tag{8}$$

In the frequency domain, the power spectrum of the received signal is expressed as

$$|Y(f)|^2 = |X(f)|^2 \cdot |H(f)|^2, \tag{9}$$

where the reverberation spectrum $H(f)$ is typically frequency-dependent, leading to signal amplification or attenuation in specific frequency bands. After processing through the Mel filter bank, the Mel-spectral energy is calculated as

$$E_m = \sum_f |X(f)|^2 \cdot |H(f)|^2 \cdot Mel\,(f). \tag{10}$$

Reverberation alters the energy distribution across Mel-frequency bands. Following logarithmic compression, the smoothing effect and energy redistribution in the spectrum are further amplified, ultimately causing deviations in MFCC coefficients from their original values.

Thus, although the MFCC method exhibits moderate robustness in anti-interference capability—effectively processing weakly disturbed acoustic signals to retain feature similarity with the original target signals, thereby facilitating model discrimination between sample classes—its extracted features remain significantly compromised under strong interference conditions. Substantial deviations in sample features increase the likelihood of target misclassification and degrade model learning efficacy.

B. Underwater acoustic signal sample distribution

Based on the preceding discussion, it is evident that underwater acoustic data acquisition is subject to varying degrees of interference. Although the MFCC method exhibits moderate robustness and can withstand certain interference levels, it fails to perform effectively on data contaminated by strong interference. When such data are included as model samples, the extracted features deviate from the expected target characteristics, failing to represent genuine target attributes. These samples, which may introduce model interference and degrade performance, are empirically termed *anomalous samples*, whereas unaffected samples are classified as *normal samples*. Anomalous samples refer to data points exhibiting pronounced feature deviations relative to other samples within the same category, whereas normal samples exhibit minimal deviation values relative to their counterparts.

Consequently, the collection of target acoustic data in underwater environments presents inherent difficulties and unavoidable challenges. Compared to terrestrial data acquisition, effective underwater acoustic datasets are significantly fewer in quantity and contain a higher proportion of contaminated data. Thus, target recognition algorithms for underwater acoustic signals must account for both limited data availability and suboptimal data quality.

Furthermore, in deep learning frameworks, small-sample datasets often suffer from inherent class imbalance, posing challenges to learning efficacy. Such imbalance typically stems from systematic biases during data acquisition, which may arise from collector preferences, source limitations, or methodological imperfections in data collection. Additionally, natural distribution imbalances also contribute significantly to class disproportion in small-sample scenarios.[29]

In our subsequent discussion, we assume that the existing underwater acoustic small-sample training set X contains a total of N labeled samples with recorded sample labels, namely $X = x_1, x_2, x_3, \ldots, x_N$. The anomalous sample rate in the training set is denoted as $\lambda\,(\lambda \in (0, 1))$, which is treated as an abstract contamination ratio rather than a precisely measurable random variable. The parameter λ is influenced by factors such as sample collection time, environmental conditions, and instrumentation and is therefore subject to inherent variability in practical data acquisition. Consequently, the training set comprises $\lambda \cdot N$ anomalous samples and $(1 - \lambda) \cdot N$ normal samples.

IV. THE PROPOSED SUDG

In this section, we elaborate on the proposed SUDG. The discussion is organized into three components: 1) the forward propagation method based on data grouping; 2) the backpropagation method incorporating inter-group weight adjustment; and 3) the group selection and dataset allocation strategy. The overall workflow of the algorithm is illustrated in Fig. 1. Detailed explanations of these components follow.

A. Forward propagation method based on data grouping

The forward propagation method based on data grouping is illustrated in Fig. 2. In the forward propagation phase, to mitigate the performance interference caused by anomalous samples on the acoustic recognition model, we introduce a training set grouping strategy prior to model training. Given the limited size of the training set X, directly partitioning it into n groups would result in each subset containing only N/n samples, drastically reducing the data volume per group. To address this, an interleaved grouping strategy is applied to large yellow croaker sound training set. Because the quality of samples cannot be predetermined, all $x_1, x_2, x_3, \ldots, x_N$ samples are first randomly shuffled before training. Subsequently, the shuffled samples are evenly partitioned into n splitting-datasets, denoted as $\{Q_1, Q_2, Q_3, \ldots, Q_n\}$.

https://doi.org/10.1121/10.0043129

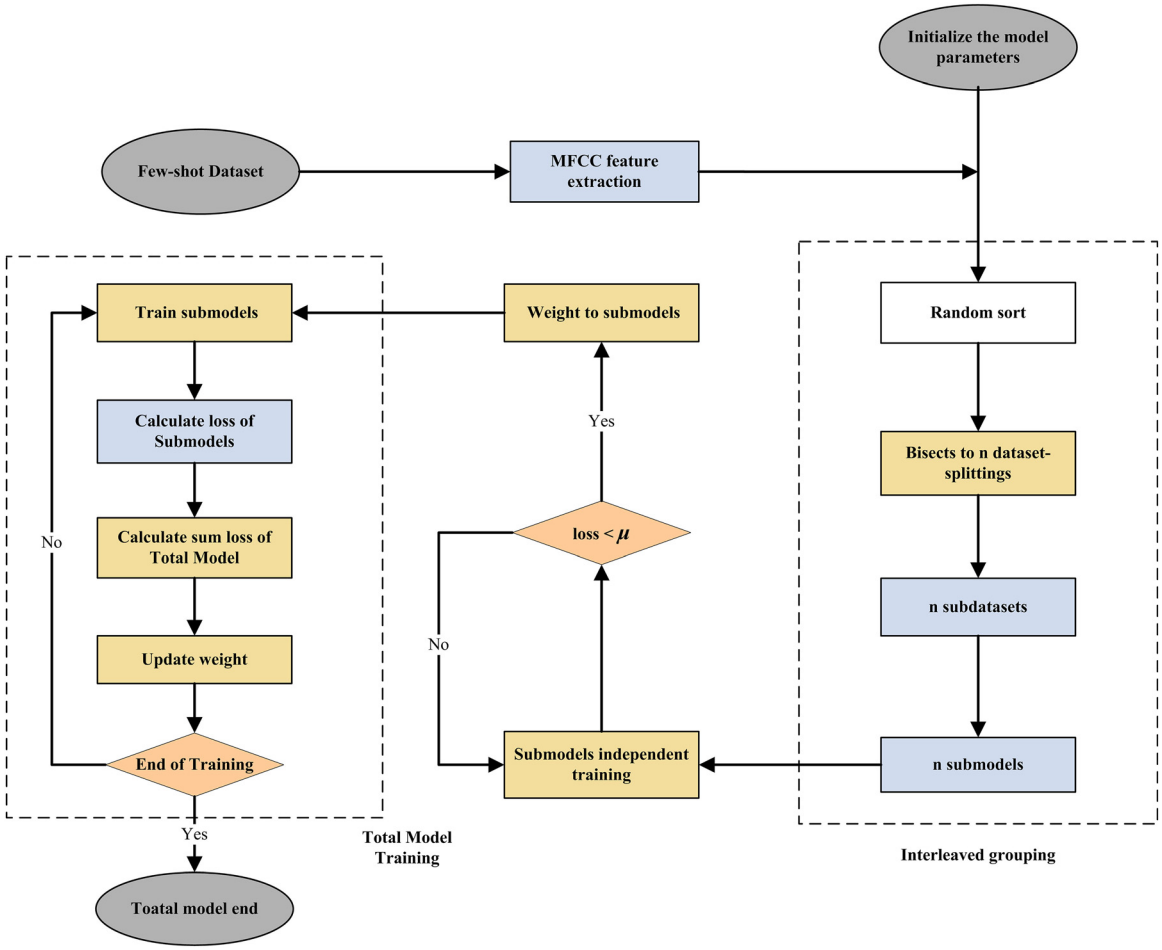

FIG. 1. Algorithm flow chart of proposed SUDG.

Each splitting-dataset contains N/n samples, and all subsets are mutually exclusive with no overlapping samples. The composition of the splitting-datasets is structured as follows:

$$Q_j = \left\{x_{N*j/n-N/n+1}, x_{N*j/n-N/n+2}, \ldots\ldots, x_{N*j/n-1}, x_{N*j/n}\right\}. \tag{11}$$

Merge the splitting-datasets according to the following method:

$$T_j = X\ Q_j = \left\{x \in X \,|\, x \notin Q_j\right\}. \tag{12}$$

It forms the sub-dataset $\{T_1, T_2, T_3, \ldots, T_n\}$. The above equation indicates that the sub-dataset T_j encompasses all the splitting-datasets Q_1,Q_2,…,Q_{j-1},Q_{j+1},…,Q_n except for Q_j. Each sub-dataset incorporates $n-1$ groups of splitting-datasets, and there are no two completely identical sub-datasets. Each sub-dataset contains a total of $N - N/n$ samples.

B. Backward propagation method for weight adjustment between groups

The backward propagation method for weight adjustment between groups is illustrated in Fig. 3. After grouping the training sample sets in SUDG, the model requires training to identify sub-datasets containing anomalous samples and reduce their weights, thereby minimizing the interference of such samples on model performance. To achieve this, we integrate an advanced weight update mechanism into the training process.

Employing supervised learning, model training is performed on each of the n sub-datasets obtained in the previous step, yielding sub-models $M_1, M_2, M_3, \ldots, M_n$ corresponding to the sub-datasets $T_1, T_2, T_3, \ldots, T_n$. Each sub-model M_k $(k = 1, 2, \ldots, n)$ is assigned a weight a_k, with identical initial weights.

All sub-models undergo multiple cycles of model training, during which the loss value for each training iteration is computed and recorded. Sub-models that undergo continuous σ rounds of training and satisfy the condition that the loss value remains below μ are incorporated into the final ensemble model.

Assuming there are m sub-models satisfying the loss value condition, these are reindexed as M_k $(k = 1, 2, \ldots, m)$. The global model W, defined as the ensemble of all qualified sub-models, is responsible for computing the aggregate loss function *Loss*, updating the weights a_k of the m sub-models, and generating the final prediction *OUTPUT* for sample classification. Sub-models are synchronously trained under W,

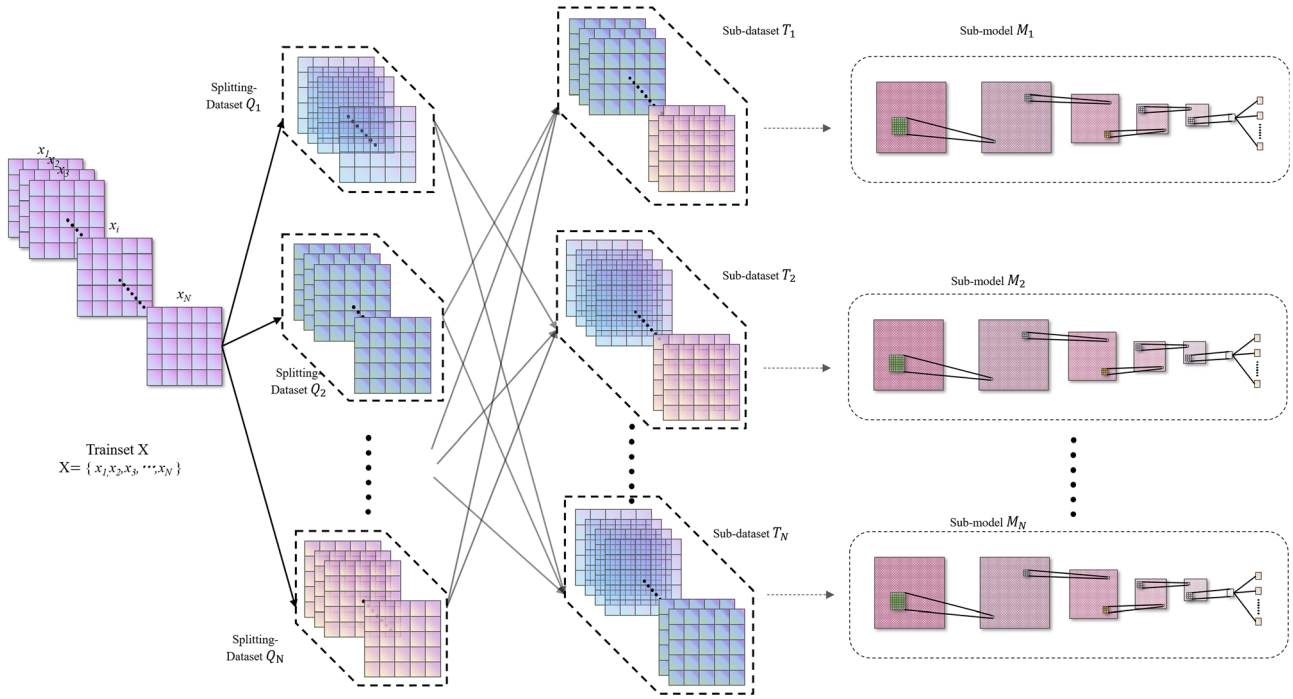

FIG. 2. Schematic diagram of forward propagation method based on data grouping.

meaning that all sub-models independently complete one training epoch concurrently with each global epoch of W, yielding the per-epoch loss value l_k for each sub-model M_k.

The total loss function *Loss* at the epoch-th training iteration is computed as follows, based on the loss values l_k and weights a_k of the sub-models M_k,

$$\text{Loss}_{epoch} = \frac{\sum_{k=1}^{m} a_k l_k}{\sum_{k=1}^{m} a_k}. \tag{13}$$

According to the gradient descent algorithm, compute the total loss function *Loss* of the global model and its partial derivatives with respect to each weight a_k. Set the learning rate γ for gradient descent and compute the gradient descent direction for the weights a_k. The partial derivative formula is expressed as

$$\frac{\partial Loss}{\partial a_k} = \frac{\sum_{i=1}^{m} a_i}{\left(\sum_{i=1}^{m} a_i\right)^2} l_k + \frac{-a_1}{\left(\sum_{i=1}^{m} a_i\right)^2} l_1 + \frac{-a_2}{\left(\sum_{i=1}^{m} a_i\right)^2} l_2 + \frac{-a_3}{\left(\sum_{i=1}^{m} a_i\right)^2} l_3 + \cdots + \frac{-a_m}{\left(\sum_{i=1}^{m} a_i\right)^2} l_m. \tag{14}$$

After each training epoch, the sub-model weights a_k are updated according to the following rule:

$$a_k = a_k - \frac{\partial Loss}{\partial a_k} * \gamma. \tag{15}$$

After training for E epochs, the training of the global model W is completed.

To compute the output of the global model W, it is required to compute the output value $output_k$ of each sub-model for a single sample. The total predicted output of the model during each testing iteration is obtained by performing a weighted summation of the outputs from all sub-models

$$\boldsymbol{OUTPUT} = \frac{\sum_{i=1}^{m} a_i * \boldsymbol{output_i}}{\sum_{i=1}^{m} a_i}, \tag{16}$$

where $\boldsymbol{OUTPUT} \in \mathrm{R}^{\mathrm{C}}$ denotes the final output logits of the model for each of the C classes. To interpret these values as class probabilities, the softmax function $\sigma(\cdot)$ is applied,

$$\boldsymbol{P} = \sigma(\boldsymbol{O}) = [p1, p2, \ldots, p_C], \tag{17}$$

where

$$p_i = \exp(\boldsymbol{O_i}) \Big/ \sum_{j=1}^{C} \exp(\boldsymbol{O_j}). \tag{18}$$

The predicted class label $\hat{y}$ is then assigned by selecting the class with the highest probability

$$\hat{y} = argmax(\boldsymbol{P}). \tag{19}$$

Therefore, the global prediction output ***OUTPUT*** is converted into binary values (0/1), yielding the predicted classes of the test set, which reveals the final test results.

https://doi.org/10.1121/10.0043129

To illustrate the dataset partitioning strategy more clearly, we present the pseudocode in Algorithm 1.

ALGORITHM 1. Acoustic target recognition via small-sample deep learning with a proposed dataset partitioning strategy.

Input: Dataset
Output: Classification model
Initialization: Model parameters, random sort dataset
1: Calculate MFCC feature extraction of dataset
2: Bisects to n dataset-splittings
3: Combine to n subdatasets
4: For subdatasets = 1, 2, …, *n*, **do**
5: Train submodels by subdatasets
6: **For** epoch = 1, 2, …, *E* **do**
7: Train model
8: Test model
9: Calculate *loss*
10: **If** *loss* < μ **then**
11: submodels ← weight *a*
12: Total model ← submodels
13: break
14: **end if**
15: **end for**
16: end for
17: For epoch = 1, 2, …, *E* **do**
18: **For** subdatasets = 1, 2, …,*n*, **do**
19: Train submodels by subdatasets
20: Train model
21: Test model
22: Calculate *loss*
23: **end for**
24: Calculate *loss* of total model
25: Update weight *a*
26: **end for**

The definition of μ centers on the "basic assessment of model effectiveness in few-shot scenarios." Its primary objective is to screen base models that exhibit preliminary learning capability and stable performance, thereby preventing ineffective models from introducing noise and interfering with joint training. Essentially, μ serves as a "statistical baseline threshold for the initial performance of base models," derived from the following two principles:

The first principle is the theoretical lower-bound constraint, which aims to exclude models performing at the level of random guessing. Taking the large yellow croaker classification task as an example, for a C-class classification problem, the baseline accuracy for random guessing is 1/C. μ must exceed this theoretical lower bound to ensure that the selected models have learned task-relevant features rather than merely generating random outputs.

The second principle is the statistical stability constraint. Due to the inherent variability in few-shot training, μ should be determined based on statistical results from multiple initial training experiments. This helps avoid incorrectly discarding valid models or including unstable ones due to the randomness of a single training run.

Considering that different sub-models are trained on different data, which may lead to inconsistent performance and affect unified conclusions, we adjusted the method for setting μ.

Under the premise of maintaining consistent training parameters across all sub-models, to ensure the reproducibility of the experiment, we have designed the threshold parameter μ as follows:

μ employs a fixed threshold based on the theoretical lower bound, which is defined as

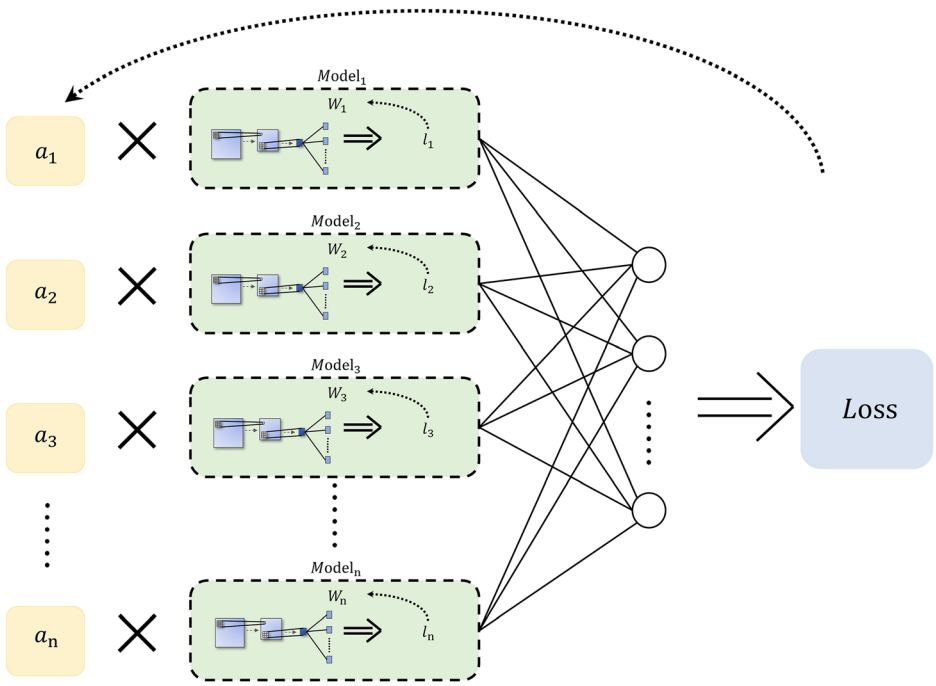

FIG. 3. Diagram of backward propagation method.

$$\mu_{base} = \frac{max\{n_1, n_2, \ldots, n_C\}}{N_{Test}} + \delta, \tag{20}$$

where C is the number of categories, the sample size for each category is n_i ($i = 1, 2, \ldots, C$), the total sample size is

$$N_{Test} = \sum_{i=1}^{C} n_i, \tag{21}$$

and δ is a small positive number (set to the minimum change in the test set in the experiments, $1/94 = 0.0106$). This threshold is directly derived from the theoretical random guessing baseline 1/C for classification problems. This threshold is determined solely by the class structure of the dataset and is computed deterministically without the need for any pre-training or distributional assumptions, ensuring the complete reproducibility of the method.

To address the high variance in performance inherent in few-shot learning, we further introduce a stability assessment mechanism. Each candidate sub-model undergoes 50 independent pre-training runs and is selected for subsequent joint training only if it meets or exceeds the above fixed threshold in a sufficient proportion of the runs. This mechanism aims to screen for sub-models that are both stable and excellent in performance, thereby enhancing the efficiency and robustness of the subsequent ensemble training.

We have verified the stability of this design through systematic sensitivity analysis (detailed in Appendix). The results show that when δ is within the range of 0.01–0.03 and the passing proportion is within the interval of 60%–80%, the performance fluctuation of the final joint model is less than 3%, indicating that the method exhibits good robustness to parameter choices. This configuration can effectively eliminate abnormal models with low performance or excessive fluctuations.

C. Number of groups and dataset partitioning method

In SUDG, the selection of the group quantity n critically determines the training volume of each sub-model, thereby influencing the overall model performance. For instance, if $N = 250$, setting $n = 2$ results in sub-dataset training sets $T1 = T2 = 125$, which drastically reduces the sample size per sub-model. Conversely, a large n value maintains near-original sample sizes in each splitting-dataset but escalates computational overhead. Specifically, when $n = N$, each splitting-dataset $\{Q_1, Q_2, Q_3, \ldots, Q_N\}$ contains only one sample, enabling precise quality assessment of individual samples within sub-models. However, this configuration increases the global model's total training volume to $N*(N\text{-}1)*epoch$, doubling the computational complexity compared to the non-grouped baseline $N*epoch$.

Therefore, selecting an appropriate data splitting quantity n significantly impacts model performance. Based on the above analysis, we tentatively assume $2 < n < N$. Given the anomalous sample quantity λN, the range of maximum anomalous sample count N_0 among the n splitting-datasets is

$$\begin{cases} 1 \le N_0 \le \dfrac{N}{n}, & n < \lambda N, \\ \lambda\dfrac{N}{n} \le N_0 \le \dfrac{N}{n}, & kn = \lambda N, \\ \lambda\dfrac{N}{n} < N_0 \le \dfrac{N}{n}, & kn > \lambda N, \end{cases} \tag{22}$$

where $k = 1, 2, 3, \ldots$. The highest value of the anomalous sample rate $\lambda_0 = N_0/N/n$ among all groups is

$$\begin{cases} \lambda_0 \ge \dfrac{n}{N}, & n < \lambda N, \\ \lambda_0 \ge \lambda, & kn = \lambda N, \\ \lambda_0 > \lambda, & kn > \lambda N. \end{cases} \tag{23}$$

We can observe that when $kn \neq \lambda N (k = 1, 2, 3, \ldots)$, at least one sub-training set among the n splitting-datasets will inevitably contain an anomalous sample quantity greater than $\lambda \times N/n$.

And when $kn = \lambda N (k = 1, 2, 3, \ldots)$, it follows that $\lambda_0 \ge \lambda$. Therefore, the proportion of grouping configurations satisfying $\lambda_0 > \lambda$ can be expressed as $1 - P(\lambda_0 = \lambda)$.

To compute $P(\lambda_0 = \lambda)$, when $\lambda N \le N/n$, the anomalous samples are approximately distributed across groups. The total number of possible distributions across n groups is n^{kn}. The number of configurations where each group contains exactly k anomalous samples is given by $(kn)!/(k!)^n$. Thus, the probability is

$$P(n, k) = \frac{(kn)!}{(k!)^n n^{kn}}. \tag{24}$$

Similarly, for $k + 1$, we have

$$P(n, k+1) = \frac{(kn+n)!}{[(k+1)!]^n n^{(k+1)n}}. \tag{25}$$

The ratio between $P(n, k+1)$ and $P(n, k)$ is calculated as

$$\frac{P(n, k+1)}{P(n, k)} = \frac{(kn+n)\cdot(kn+n-1)\cdots(kn+1)}{[(k+1)\cdot n]^n}. \tag{26}$$

Because $n \ge 3$, this ratio is less than 1, implying $P(n, k+1) < P(n, k)$.

Consequently, $P(n, k)$ is maximized when $k = 1$, yielding

$$P(n, 1) = \frac{n!}{n^n}. \tag{27}$$

For $k = 1$ and $n = 3$, $P(n, k)$ reaches its maximum value, leading to $P(\lambda_0 = \lambda) \le 2/9$.

https://doi.org/10.1121/10.0043129

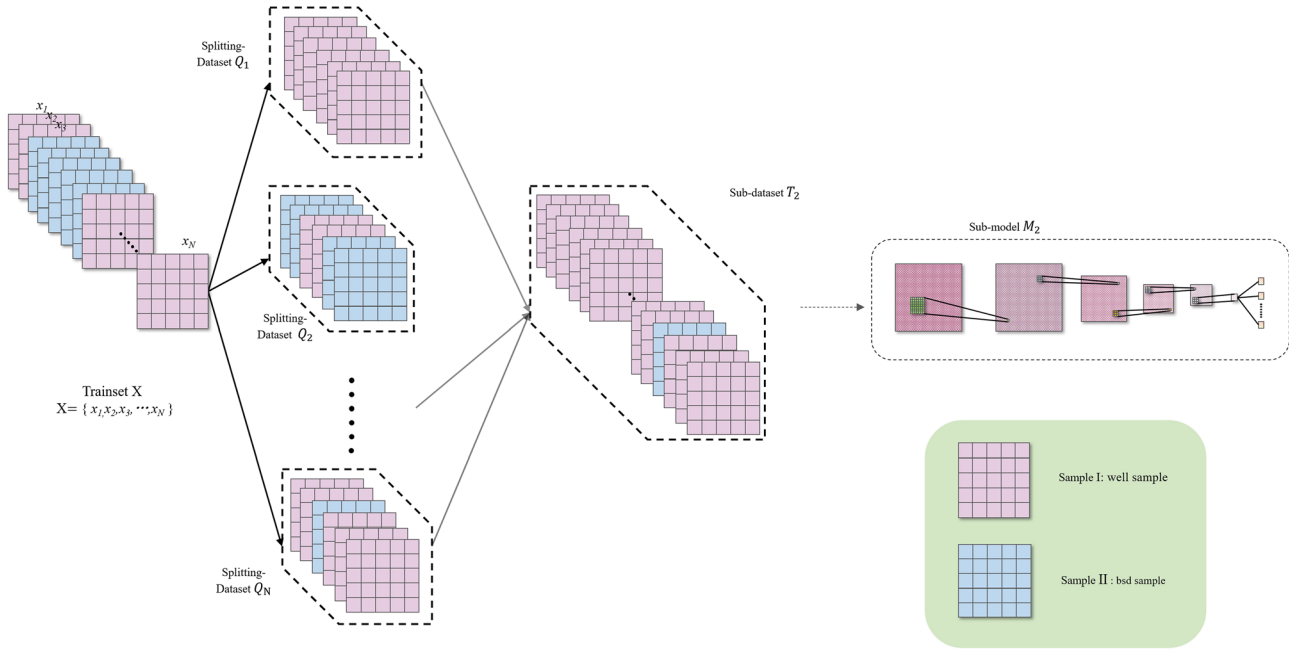

FIG. 4. Grouping effect diagram.

In the case where $\lambda N > N/n$, the distribution of anomalous samples is constrained by the group capacity limits. This can be analyzed based on the distribution of normal samples, and similarly, we obtain $P(\lambda_0 = \lambda) \leq 2/9$. Hence, the proportion of grouping configurations satisfying $\lambda_0 > \lambda$ is bounded below by 7/9.

This indicates that, in most cases, the grouping process generates a splitting-dataset with an anomalous sample rate $\lambda_0 > \lambda$ higher than that of the original training set. Taking Fig. 4 as an example, since subgroup Q_2 exhibits the highest anomalous sample rate, cross-group partitioning will produce a sub-training set T_2 excluding Q_2. The anomalous sample rate of sub-training set T_2 satisfies $\lambda_0 < \lambda$. After deep learning training, this sub-training set generates a sub-model M_2, which achieves superior performance. Subsequent weight adjustments in the global model further enhance the effectiveness of M_2, ultimately resulting in a global model with improved performance compared to scenarios without grouping. Therefore, this method tends to achieve improved model performance in the presence of anomalous samples when $\lambda \in (0, 1)$.

In the classification task of a large-sample balanced dataset, the dataset is typically divided into a training set and a test set in a 7:3 ratio,[30] without intervening in the proportions of each category, so the ratios of each category remain consistent with those in the dataset. However, the same division method cannot achieve reasonable results in small-sample classification tasks.

In our work, assuming the proportion of tail classes to all categories is η, the total number of tail class samples is ηN. Thus, the ratio of tail class samples to anomalous samples in the training set is

$$\frac{\eta(1-\lambda)N}{\lambda N} = \frac{\eta(1-\lambda)}{\lambda}. \tag{28}$$

This implies that as the anomalous sample rate λ increases, the quantity of tail classes becomes closer to that of anomalous samples, and both tail classes and anomalous samples may lead to confusion in the model. Consequently, proper partitioning of the dataset is essential under small-sample conditions. In Section V A, we conducted experiments comparing the performance between large-sample datasets and small-sample datasets under identical class proportions, without implementing dataset balancing allocation.

The theoretical analysis in this section assumes independent random sampling. In practice, we approximate this condition by randomly shuffling the raw time series prior to analysis, which disrupts sequential order and mitigates short-range temporal dependencies. Although the raw data may exhibit temporal correlations, the shuffled data align more closely with the independence assumption underlying the theoretical bounds. The derived probability upper bound, therefore, serves as a theoretical reference for method robustness under randomization, not as an absolute guarantee for unprocessed time series. Rather, it provides a principled baseline for empirical comparison when randomization is applied.

We acknowledge that random shuffling, though substantially reducing temporal correlations, does not eliminate all inter-sample dependencies. Complex structures, such as long-range correlations, latent periodicities, or non-stationarities, may persist. Accordingly, the bound should be regarded as offering directional insights and a design-phase reference, not a rigorous certificate for all possible data configurations.

V. DISCUSSION OF EXPERIMENTAL RESULTS

A. Difference and comparison between large and small samples

In this study, we employed the publicly available audio classification dataset UrbanSound8K primarily as a methodological benchmark to evaluate the proposed SSL framework under controlled conditions.

The UrbanSound8k dataset consists of 8732 labeled urban sound clips spanning ten distinct categories, such as dog barking, drilling, and sirens.[31] This dataset has been widely adopted in general audio classification tasks, due to its diverse categories and substantial sample size.[32,33] We explicitly note that the acoustic propagation characteristics (e.g., channel effects, attenuation) in this airborne urban sound dataset differ fundamentally from underwater bioacoustics environments. Its use here is strictly for assessing algorithmic robustness and generalization in a data-limited setting, not for validating physical acoustic properties relevant to underwater scenarios.

To investigate the performance of small-sample datasets in model training and validate their high sensitivity to data allocation, we designed a series of experiments using this benchmark dataset. In these experiments, we established both large-sample and small-sample datasets while maintaining their class distributions identical to the original dataset. Specifically, we deliberately avoided any intervention in the class distributions, ensuring consistent class proportions across the training set, test set, and original dataset. This experimental design aims to eliminate potential biases caused by class distribution discrepancies, thereby enabling a more accurate evaluation of sample size effects on model performance in a general audio classification context.

In the experimental setup, we utilized three categories of audio clips to construct training and testing set ratios ranging from 1:9 to 9:1, comprising multiple experimental groups. This configuration allows systematic investigation of the impact of varying training set sizes on model performance, particularly under small-sample conditions. The dataset exhibits an imbalanced class distribution of 100:50:84, indicating unequal sample quantities across categories. Furthermore, the large-sample (L-sample) dataset contains five times the sample quantity of the small-sample (S-sample) dataset, amplifying the influence of sample size on model training.

This experimental design enables comprehensive evaluation of model sensitivity to dataset quantity allocation under different sample size conditions, with particular emphasis on small-sample dataset performance. Such analysis primarily provides methodological insights into the relationship between sample size and model effectiveness in a controlled setting, and its direct implications for underwater bioacoustics are limited by the dataset's domain mismatch.

The experiments employed the F1-score as the evaluation metric. The F1-score is a comprehensive metric for assessing classification model performance, which harmonizes precision (the ratio of true positives to predicted positives) and recall (the ratio of true positives to actual positives) by calculating their harmonic mean. This balance makes it particularly suitable for class-imbalanced scenarios. The F1-score is computed as

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}. \tag{29}$$

Precision measures the accuracy of the model's positive class predictions, whereas recall quantifies its ability to identify true positive instances. The F1-score ranges from 0 to 1, with values approaching 1 indicating superior model performance. By integrating both precision and recall, the F1-score provides a comprehensive evaluation of classification efficacy, making it particularly valuable in applications, such as medical diagnosis and text classification, where class imbalance or operational trade-offs are critical considerations.

FIG. 5. Comparison of F1-scores of different categories under different proportions.

The results are shown in Fig. 5. Class 1, with the lowest proportion in the dataset, is identified as the tail class. In the small-sample dataset, the sample size of Class 1 is only 50—half that of Class 0—and the training set contains even fewer samples. This leads to a significant weakness in the model's ability to recognize Class 1 features, resulting in an F1-score of 0.0% across proportions ranging from 10% to 80%. Only at the 90% proportion does it achieve its best F1-score of 50.0%. In contrast, the model trained on the large-sample dataset maintains stable F1-scores above 80.0% for all classes, showing no obvious class imbalance. This demonstrates that without intervention in the dataset proportions, the model lacks class balance in this specific UrbanSound8K experimental setup.

https://doi.org/10.1121/10.0043129

TABLE I. Classification accuracy (%) of different methods across various training set proportions.

	10%	20%	30%	40%	50%	60%	70%	80%	90%
S-sample (SUDG)	55.6%	81.4%	84.0%	92.9%	88.9%	92.5%	92.8%	97.9%	100.0%
L-sample (baseline)	88.3%	93.5%	94.9%	97.0%	96.1%	97.0%	97.2%	97.4%	96.6%
S-sample (baseline)	28.5%	19.4%	25.5%	68.5%	69.3%	68.7%	69.8%	69.8%	80.8%

This occurs because, under identical ratio conditions, the large-sample dataset contains a larger number of tail class samples, enabling the model to learn sufficient features of tail classes and ensuring baseline recognition performance for these categories. However, in small-sample datasets, the limited quantity of tail class samples may lead the model to either acquire erroneous features from sparse tail class instances or misclassify tail classes as anomalous samples due to their extreme scarcity, resulting in significantly weakened recognition capability for these categories.

Consequently, tail classes exert a more substantial impact on model performance in small-sample training scenarios. This underscores the critical need to address class distribution imbalances during dataset partitioning for small-sample conditions to optimize model effectiveness. These findings are presented as general observations on data allocation for imbalanced, small-sample learning, and their transferability to underwater bioacoustics tasks would require validation with domain-specific datasets that account for the pertinent acoustic propagation physics.

B. Performance of SUDG on small-sample

Subsequently, we conducted comparative experiments between the proposed SUDG small-sample model and the large-sample baseline model using identical data types from the UrbanSound8K dataset, with the results summarized in Table I. For comparison, we also reported the performance of baseline when a small-sample dataset is available for training (denoted as S-sample). This comparison is intended to evaluate the relative efficacy of the proposed framework under data scarcity in a general audio domain.

From the analysis of this data, it can be observed that although the small-sample method demonstrates capabilities to resist interference from anomalous samples, enhance model recognition accuracy, and balance tail classes, its performance on unseen data remains insufficiently stable and reliable due to limited training data volume, resulting in a noticeable performance gap compared to the large-sample model. Specifically, the small-sample approach exhibits performance fluctuations between 81.4% and 97.9%, whereas the conventional large-sample method maintains stable performance across all proportions, consistently ranging from 88.3% to 97.4%.

The primary reason for this discrepancy lies in the small-sample method's inherent limitation in volume of data, which restricts its ability to fully capture the complex structural relationships and latent patterns within the data. Consequently, this leads to compromised generalizability and reliability when processing the data that has never been seen. In contrast, models trained on more abundant datasets can learn the data distribution and feature representations more comprehensively, thereby enhancing their upper-bound performance and demonstrating superior robustness in most operational scenarios. We emphasize that the conclusions drawn from this UrbanSound8K-based experiment are confined to the context of the dataset's characteristics. They serve to illustrate the challenges and behavior of SSL in audio classification but do not directly substantiate claims about robustness in underwater acoustic environments, where channel effects and attenuation present distinct challenges.

C. Data acquisition and preprocessing

The dataset used in this study was collected on April 11, 2024, at the offshore laboratory of Xiamen University for the purpose of gathering vocalization data from large yellow croakers. The experiment was conducted in a controlled laboratory aquarium with dimensions of 65 × 30 × 45 cm and a water volume of approximately 75 L. We used six healthy large yellow croakers as sound-producing subjects. The data were collected through sequential individual monitoring—each fish was placed alone in the aquarium and recorded using an ullra-low power acoustic signal acquisilion system (Hangzhou SONICINFO Technology Co. Ltd., Hangzhou, ZheJiang, China) self-contained acoustic recorder to accurately associate vocalization events with specific individuals. Recordings were made under very low light conditions to minimize stress on the fish. The raw audio data consisted of two files: 200 A_240410221500_16384_20_00000.bin and 200 A_240410224820_16384_20_00001.bin, with a total duration of 66 min and 19 s (approximately 66.32 min). The sampling rate was 16.384 kHz, and the frequency response ranged from 20 Hz to 20 kHz (within ±1.5 dB fluctuation), ensuring the integrity of the data within the human audible frequency band and making it suitable for analyzing large yellow croaker vocalizations. The original data were acquired using the LoPAS-L self-contained acoustic recorder, and the raw bin files were converted to wav format using LoPAS-L (v1.1) software.

The LoPAS-L recorder integrates a complete signal conditioning circuit, with specific processes and parameters as follows: it includes a low-noise preamplifier with a gain of 20 dB, which is essential for amplifying the weak voltage signals from the hydrophone to a level suitable for digitization, thereby effectively ensuring a high SNR. The recorder also incorporates a sharp-cutoff anti-aliasing filter. Based on our sampling rate of 16.384 kHz, the filter's cutoff frequency is set at approximately 8.192 kHz, strictly adhering

to the Nyquist sampling theorem, which ensures no aliasing distortion within the effective analysis frequency band of 0–8 kHz. In addition, the hydrophone itself has a frequency response of 20 Hz–20 kHz (±1.5 dB) and a high sensitivity of –192.6 dB. Together, this system ensures high-quality capture of the bioacoustics signals of large yellow croakers. Therefore, our claim of high suitability is based on the recorder's high sensitivity, wide flat frequency response, and the professional signal conditioning chain—including 20 dB preamplification and anti-aliasing filtering—making it particularly suitable for capturing and recording croaker bioacoustics signals.

In the data preprocessing stage, obvious noise was excluded and large yellow croaker vocalization signals were screened. Specifically, we extracted individual pulse signals of large yellow croakers from the original audio as individual samples through manual inspection and an algorithm based on signal similarity and peak detection. Each sample is a short audio segment of 10 ms, ultimately forming a dataset for training and testing. The entire dataset contains approximately 234 samples, ensuring representativeness and diversity. The data processing workflow is designed to extract meaningful features from the original audio for use in machine learning models. The specific steps are as follows: Signal screening and segmentation: First, we preprocess the original audio using a pulse detection algorithm based on stable region detection to identify and extract individual pulse signals of large yellow croakers. Each pulse signal is segmented into fixed 10 ms segments to maintain data consistency. Next, feature extraction is performed on each 10 ms audio segment. MFCC processing is employed to convert the time-domain signal into a frequency-domain representation. Specifically, we computed 13 MFCC coefficients and further generated Mel spectrograms as input to the model. Mel features effectively simulate the human ear's perception of frequency and preserve key patterns in acoustic signals, making them particularly suitable for fish vocalization classification tasks. The input to the model is the Mel spectrogram, with dimensions of 1×128. These spectrograms are normalized and converted to a logarithmic scale to enhance feature discriminability. This processing workflow ensures that the data retains bioacoustics characteristics while reducing noise interference and provides standardized input for subsequent model training.

Finally, the sampling rate was explicitly set to 16.384 kHz, providing an effective analysis bandwidth of 0–8.192 kHz, which fully covers the typical energy concentration range of croaker vocalizations, especially the spectral peak around 400–600 Hz. The classification in this study is based on individual fish: Class 0, Class 1, and Class 2 correspond to three different large yellow croaker individuals. From the total dataset of six fish, we selected the three individuals with the highest vocalization quality and most sufficient data for this classification study. In terms of sample size and handling of anomalous samples: through a combination of manual and algorithmic screening, we extracted a total of 234 clean croaker pulse signals as valid samples from the original recordings. During the data collection phase, anomalous samples caused by environmental noise or fish collisions—i.e., non-target sound sources or noise—were proactively identified and excluded during preprocessing through visual inspection and a similarity- and peak-detection-based algorithm. The final dataset of 234 samples consists of confirmed valid croaker vocalizations, with each sample being a 10-ms audio segment.

In the experiments, we employed several representative deep learning architectures, including ResNet34, MobileNet_v2, and Convolutional Neural Network–Long Short-Term Memory (CNN–LSTM) network. These models are widely used in computer vision and sequence modeling tasks and have been successfully applied to the field of acoustic signal analysis. A brief description of the specific models is as follows: ResNet34: As a residual network, it addresses the gradient vanishing problem in deep networks through skip connections, making it suitable for learning complex acoustic features. MobileNet_v2: This is a lightweight network that uses depthwise separable convolutions to reduce computational costs, making it suitable for resource-constrained environments. CNN-LSTM: This model combines the spatial feature extraction capability of CNN with the temporal sequence modeling ability of LSTM, making it suitable for spatiotemporal patterns in acoustic signals. The selection of these models was based on their performance in large yellow croaker sound classification, and we compared their performance through experiments, as shown in Table IV. Furthermore, the network parameter configuration is specified in Table II, and the detailed numbers of training and test samples are clearly listed in the revised Table III.

The SNR was computed as a power-based ratio between the target signal segment and the surrounding background noise. To ensure reliable annotation and model training, recordings with an average SNR below 5 dB were excluded,

TABLE II. Simulation parameter settings.

Project	Details	Simulation parameters	Value
Task Type	Closed set recognition	Maximum epoch (E)	100
Total number of vocalization events	74	Number of groups (n)	3,4,5
Total number of pulses	234	The sample size of Epoch	234
Total number of individuals	3	Learning rate (γ)	0.001
Data partition unit	Independent event	Threshold (μ)	0.5425
Partition strategy	Time blocks retained	Num class	3
Trainable parameters	23 63 532	Input channel	1
Optimizer	Adam	Batch	32

https://doi.org/10.1121/10.0043129

TABLE III. Class definition.

Category	Scientific name	Individual ID	Samples
Large yellow croaker	*Larimichthys crocea*	Lc1	100
Large yellow croaker	*Larimichthys crocea*	Lc2	50
Large yellow croaker	*Larimichthys crocea*	Lc3	84

as such samples are typically dominated by background noise and do not contain clearly identifiable acoustic events.

For the remaining dataset (shown in Fig. 6), the SNR values span a wide range (approximately 10–60 dB). Higher SNR values correspond to impulsive vocalizations recorded under low background noise conditions or at close proximity to the sensor, which is common in passive acoustic monitoring scenarios. No artificial denoising or amplitude enhancement was applied during preprocessing.

In this experiment, we combined the prediction results of multiple models of the same architecture to obtain a model result that highly utilizes the data. We independently trained and evaluated the three models, ResNet34, MobileNet_v2, and CNN-LSTM, to compare their individual performances. This setup allowed us to analyze the strengths and limitations of each architecture in the task of large yellow croaker vocalization classification. During the training process, all models used the same dataset and preprocessing pipeline.

D. Experimental results and analysis

First, the model framework was constructed by initializing the weights of each network layer. The weights were randomly assigned following a Gaussian distribution with a mean of 0 and a variance of $\sqrt{2/fan_{in}}$, where fan_{in} represents the number of input neurons in the corresponding layer.

To investigate the role of the proposed SUDG, the impact of group quantity in the method, and its performance compared to existing classification approaches, experiments were conducted under identical small-sample datasets and data volumes. We evaluated the performance of non-grouped ResNet-34, CNN-LSTM, and MobileNet_v2 methods.

ResNet-34 is chosen due to its proven generalization capability and feature extraction advantages in both image and temporal signal processing. The initial residual structure of ResNet is also applicable in acoustic tasks. Previous studies have shown that ResNet performs exceptionally well in tasks such as speech emotion recognition and voiceprint recognition. Its moderate depth balances the model complexity and computational efficiency, enabling it to capture acoustic details while avoiding the risk of overfitting.[34,35] As a comparison benchmark, the mature architecture, reproducibility, and extensive research basis of ResNet-34 ensure its rationality as a reliable reference in cross-modal tasks.

CNN-LSTM combines convolutional neural networks and long short-term memory networks. The emergence of this hybrid model is to simultaneously capture spatial features and time series dependencies. CNN is good at handling local features and spatial information, such as edges and textures in images, whereas LSTM is good at handling time series data, such as time dependence in audio. The combination of the two is very effective when dealing with data with spatiotemporal characteristics. Literature[36,37] indicates that CNN-LSTM has a good effect on time series data.

MobileNetV2 was proposed by Sandler *et al*. from the Google team at the Computer Vision and Pattern Recognition conference in 2018.[38] Through an inverted residual structure and a linear bottleneck design, it balanced computational efficiency and feature representation capabilities in a lightweight model and quickly became the authoritative benchmark for mobile visual tasks. In recent years, this model has been widely transferred to the field of acoustic signal processing,[39–41] indicating that it has also been generally recognized in the field of acoustic tasks.

Subsequently, the best- and worst-performing methods were further analyzed by applying the grouping strategy with group quantities $n = 3,4,5$ under the same small-sample dataset and data volume. This aimed to explore how grouping affects different deep learning methods. The experimental results are summarized in Table IV. Figure 6 presents the confusion matrices of the final training results for ResNet-34, CNN-LSTM, and MobileNet_v2 without applying the SUDG method. Figures 7 and 8 display the confusion matrices of CNN-LSTM and MobileNet_v2 under the SUDG framework with $n = 3,4,5$, respectively.

1. Comparison with other algorithms

In Table IV, we primarily evaluate models using the F1-score and accuracy metrics. Among the three non-grouped

TABLE IV. Comparison of results of SUDG and other methods.

	Macro F1	acc	class0 F1	class1 F1	class2 F1
CNN-LSTM with $n = 5$	86.1% ± 2.6%	90.0% ± 1.7%	90.6%	73.3%	94.4%
CNN-LSTM with $n = 4$	87.8% ± 2.1%	90.9% ± 1.6%	91.2%	78.2%	93.9%
CNN-LSTM with $n = 3$	84.9% ± 2.1%	89.2% ± 2.5%	86.6%	60.0%	90.1%
CNN-LSTM	80.4% ± 2.8%	82.3% ± 4.4%	88.5%	18.2%	79.5%
ResNet34 with $n = 5$	80.4% ± 2.5%	83.1% ± 2.4%	84.8%	64.0%	90.1%
ResNet34 with $n = 4$	84.4% ± 2.9%	83.0% ± 2.3%	87.6%	72.7%	93.0%
ResNet34 with $n = 3$	72.5% ± 2.4%	72.9% ± 2.7%	71.6%	55.2%	84.6%
ResNet34	52.9% ± 5.0%	54.7% ± 5.7%	32.8%	28.2%	78.6%
MobileNet_v2	69.8% ± 3.7%	70.1% ± 4.5%	68.1%	60.6%	65.6%

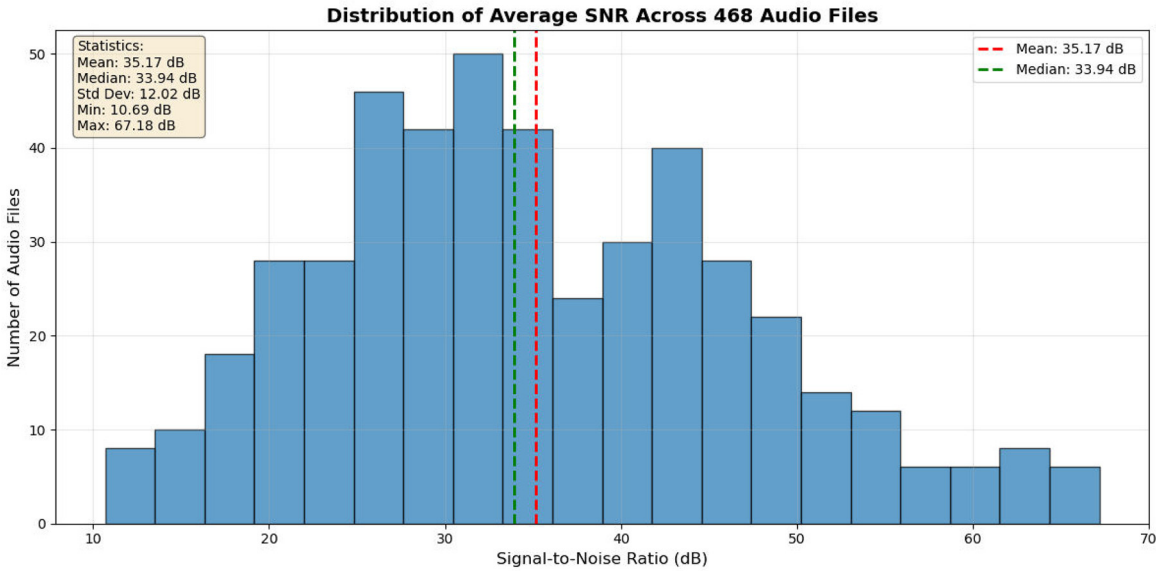

FIG. 6. Distribution of SNRs across all retained recordings.

models, ResNet34 [shown in Fig. 7(a)] demonstrates the weakest recognition performance, whereas CNN-LSTM achieves the best results [shown in Fig. 7(c)]. Although the ResNet34 method shows relatively strong recognition for certain specific categories, it generally lacks the ability to distinguish between different classes. Analysis of the confusion matrix reveals that the ResNet34 method exhibits misclassification across multiple categories, indicating deficiencies in feature extraction and class discrimination for the non-grouped ResNet34 model, which constrain its overall performance.

The MobileNet_v2 method [shown in Fig. 7(b)] demonstrates moderate effectiveness in capturing primary class features but exhibits significantly inferior performance compared to the data grouping-based small-sample recognition method. This limitation may stem from incomplete feature representation or insufficient learning of discriminative information during training, resulting in lower F1-scores and accuracy relative to the grouping-based approach.

Although the CNN-LSTM method achieves over 80% accuracy and demonstrates robust overall recognition performance, its ability to balance recognition across classes under imbalanced data conditions remains notably weaker than the data grouping-based small-sample method, particularly for tail classes. This suggests that the CNN-LSTM method may perform well on majority classes but exhibits recognition bias toward minority classes, leading to less balanced overall performance compared to the data grouping-based small-sample recognition approach.

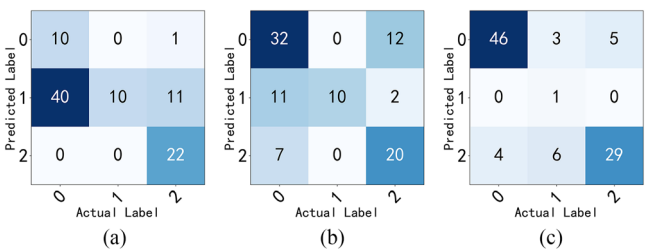

FIG. 7. Confusion matrix diagram: (a) ResNet34; (b) MobileNet_v2; (c) CNN-LSTM.

2. Comparison with the effect without SUDG method

In comparative experiments with other algorithmic models, the model with $n=4$ achieved optimal performance in both F1-score and accuracy, outperforming other methods by at least 4% in F1. The $n=5$ configuration ranked second. These results indicate that the proposed SUDG not only accurately identifies target classes but also

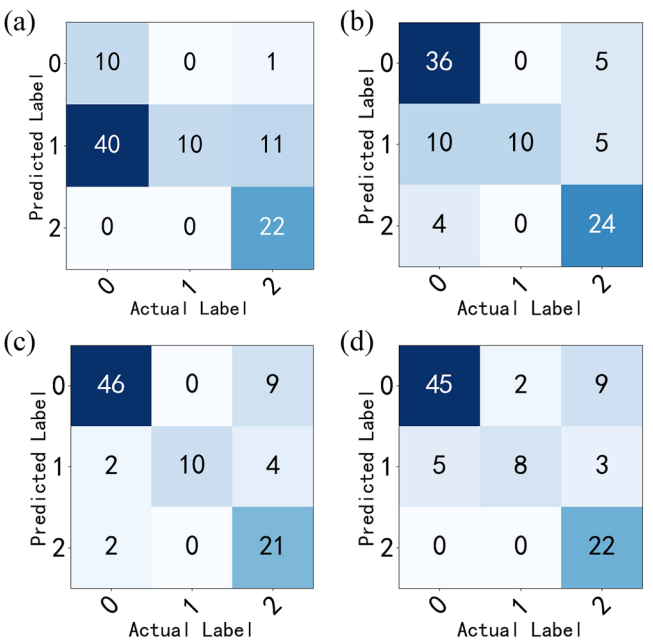

FIG. 8. Confusion matrix diagram: (a) ResNet34 model; SUDG-ResNet34 model with (b) n = 3; (c) n = 4; (d) n = 5 (n means the number of sub-models.).

https://doi.org/10.1121/10.0043129

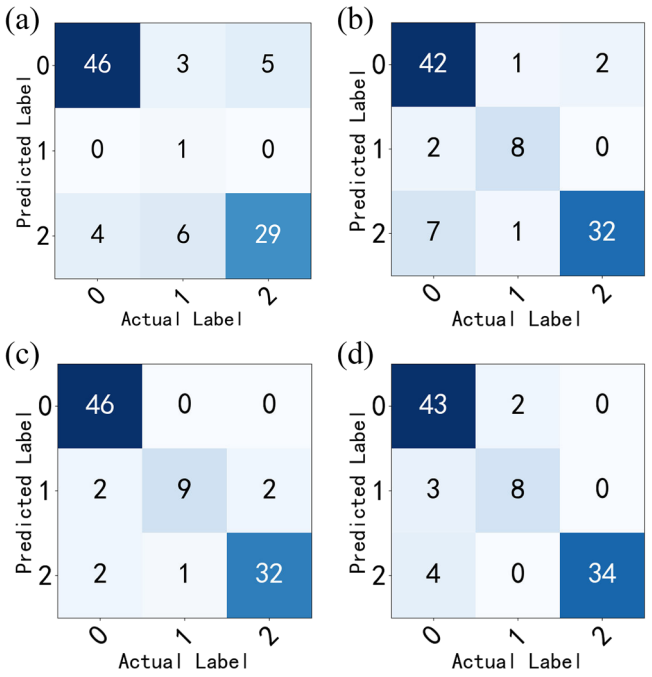

FIG. 9. Confusion matrix diagram: (a) cnn-lstm model; SUDG- cnn-lstm model with (b) n = 3; (c) n = 4; (d) n = 5 (n means the number of sub-models.).

maintains robust recognition capability under class-imbalanced conditions. The advantage of data grouping lies in its ability to balance category processing while preserving high precision and recall rates, rendering it more reliable in practical applications.

It is evident that the models employing data grouping demonstrate significant performance improvements compared to the ResNet-34 and CNN-LSTM models without the SUDG method. These grouped models successfully captured critical features in the dataset and achieved relatively accurate sample classification, with recognition accuracy increasing from 80.4% to 84.9% or higher, and the F1-score improving from 82.3% to 89.2% or above, markedly enhancing model efficacy.

Although the ResNet-34 model without SUDG exhibited decent recognition performance for certain specific classes, it lacked the ability to distinguish between different categories overall. Through analysis of the confusion matrices, we observed that this model exhibited misclassification across multiple classes, indicating deficiencies in feature extraction and class differentiation, which constrained its overall performance. Similarly, the CNN-LSTM model without SUDG demonstrated partial class-discrimination capability but showed weaker performance in identifying tail classes.

From an in-depth analysis of the confusion matrices, clear differences emerge among the four models in recognizing features within the small-sample dataset. The SUDG-enhanced models not only improved accuracy but also reduced inter-class confusion, particularly for tail classes, highlighting the method's effectiveness in addressing data imbalance and enhancing feature discriminability.

For the ResNet34 model with the weakest recognition capability, the performance improvement after grouping is substantial. The ResNet34 model without the SUDG method demonstrates notably poor performance on small-sample datasets, as shown in Fig. 8(a). In addition to its sub-50% recognition accuracy, analysis of the confusion matrix reveals that approximately 65% of its predicted labels are concentrated in Class 1. This indicates that the model largely fails to capture discriminative features across categories in the dataset and relies predominantly on random guessing during classification, reflecting an inability to effectively distinguish samples from different classes.

In contrast, the three models enhanced with the SUDG method demonstrated significant performance improvements, as shown in Figs. 8(b)–8(d). They successfully captured critical features within the dataset and achieved relatively accurate classification of samples through these features, significantly enhancing recognition accuracy.

For the CNN-LSTM model with the strongest recognition capability, although it already achieved over 80% accuracy, the SUDG method still provided a noticeable improvement. The CNN-LSTM method without SUDG, shown in Fig. 9(a), exhibited inconsistent recognition performance across categories when handling the small-sample dataset. However, as shown in Figs. 9(b) and 9(c), the grouped models better managed category weights, leading to improvements in F1-scores. Among them, the CNN-LSTM model with $n = 4$, shown in Fig. 9(c), remained the most improved, achieving an F1-score of 87%.

Under identical small-sample datasets, data volumes, and deep learning methodologies, models enhanced by the SUDG method demonstrated higher recognition accuracy compared to traditional deep learning approaches without SUDG. This confirms that applying the SUDG method can provide significant performance enhancements to the models.

3. Comparison of the effect of different number of groups

In our experiments, the small-sample deep learning UATR model with grouping quantity $n = 4$ demonstrated significantly superior performance compared to the other two models, whereas the $n = 3$ configuration exhibited poorer results. This indicates that, under identical small sample datasets, equivalent data volumes, and the same deep learning framework, varying grouping quantities critically impact model efficacy. Fewer groups lead to reduced data volume per sub-model, thereby degrading the global model's recognition performance. Conversely, excessive grouping weakens anti-interference capability. Thus, identifying an optimal grouping quantity is essential for balancing these trade-offs.

In conclusion, the proposed SUDG emerges as the most effective choice among the four evaluated algorithms, owing to its exceptional F1-scores, accuracy, and balanced recognition performance in class-imbalanced scenarios. In contrast, the non-grouped ResNet34, MobileNet_v2, and

TABLE V. Experimental results under the simple classifier.

Model	Feature type	Accuracy (%)
SVM_linear	MFCC-Spectrogram	53.2
SVM_rbf	MFCC-Spectrogram	50.0
RandomForest	MFCC-Spectrogram	53.2
KNN	MFCC-Spectrogram	47.9
LogisticRegression	MFCC-Spectrogram	45.7
NearestCentroid	MFCC-Spectrogram	44.7
CNNLSTM	MFCC-Spectrogram	80.9
SUDG	MFCC-Spectrogram	89.4

CNN-LSTM methods exhibit limitations across multiple dimensions.

E. Comparative experiments under simple classifiers

We tested the performance of large yellow croaker acoustic data on a simple classifier, as shown in Table V. The performance of all simple classifiers is confined to a narrow range (approximately 45% to 53%). This indicates that fixed handcrafted statistical features are insufficient to fully capture the discriminative information embedded in MFCC spectrograms, thereby determining the upper and lower performance bounds of such approaches. In contrast, deep learning models achieve substantially higher accuracy on the same MFCC data and test set. These models retain temporal and structural information that is typically discarded during statistical processing, allowing them to learn patterns most relevant to the classification objective.

To mitigate overfitting in the deep learning framework, we explicitly and extensively incorporated L2 weight regularization and dropout into the network architecture. The former encourages the learning of simpler patterns by penalizing excessively large weight values, whereas the latter enhances generalization by randomly deactivating a subset of neurons during training, thereby reducing reliance on specific nodes.

Furthermore, the training and test sets were kept strictly independent throughout the modeling process. This ensures that the reported performance reflects a genuine and unbiased estimate of the model's generalization capability.

F. Robustness Analysis across random seeds

To assess the robustness of the method and to rule out the possibility of overfitting to a specific random seed, we repeated all the experiments using five different random seeds. The detailed statistical results are reported in Table VI.

The calculated std and the width of the 95% confidence interval for our method are both within 3%. Given the inherent sensitivity to randomness in the small-sample setting, the observed variability across random seeds is to be expected. Factors such as model initialization and the stochastic ordering of limited data during training can have a pronounced impact on the learned representations and final decision boundaries. Meanwhile, our proposed method itself involves a stochastic component during its random grouping, which introduces an additional, legitimate source of variation across runs.

TABLE VI. Performance statistics across random seeds.

	Mean	Std	F1-95%CI	Acc-95%CI
CNN-LSTM with $n=5$	86.1%	2.6%	[0.8387, 0.8832]	[0.8851, 0.9149]
CNN-LSTM with $n=4$	87.8%	2.1%	[0.8614, 0.8974]	[0.8957, 0.9234]
CNN-LSTM with $n=3$	84.9%	2.1%	[0.8297, 0.8675]	[0.8702, 0.9107]
CNN-LSTM	80.4%	2.8%	[0.7798, 0.8261]	[0.7787, 0.8553]

Based on these, the mean performance improvement of our method over the strongest baseline consistently exceeds the observed standard deviation across runs (a gain of $>4\%$ vs a std of $<3\%$). This indicates that the improvement is stronger than the experimental variability, and the superiority of our method is statistically reliable despite the inherent stochasticity of the setting.

Therefore, we believe the reported variability is a reasonable reflection of the challenges in the SSL paradigm and does not indicate overfitting to a specific random seed. The consistent and significant mean improvement underscores the robustness and effectiveness of our approach.

G. Analysis of computing efficiency and deployment feasibility

1. Theoretical analysis of the computational cost model for ensemble methods

We first acknowledge that, from a theoretical perspective, the computational cost of training and integrating n models in SUDG is indeed higher than that of training a single model. However, this increase in cost is both manageable and predictable, ensuring that the computational overhead during both the training and inference phases remains within acceptable limits.

(1) Training phase cost

In terms of time cost, the total training time T_{train_total} exhibits an approximately linear relationship with the number of base learners n, expressed as

$$T_{train\_total} \approx n \times T_{base} + C, \tag{30}$$

where T_{base} denotes the average training time for a single sub-model, and C represents the fixed overhead associated with the ensemble operation. This linear scaling stems from the fact that the training processes of the n sub-models are mutually independent, which naturally facilitates large-scale parallel computation.

Regarding memory cost, the peak memory usage during training M_{train_peak} is primarily determined by the model complexity of a single sub-model and the batch size, i.e.,

$$M_{train\_peak} \approx M_{base}. \tag{31}$$

Because the sub-models can be trained sequentially, the memory footprint does not scale linearly with n, significantly reducing the hardware resource requirements.

(2) Inference phase cost

For inference time per sample $T_{infer\_per\_sample}$, it also scales linearly with n,

$$T_{infer\_per\_sample} \approx n \times T_{infer_{base}} + T_{ensemble}, \quad (32)$$

where T_{infer_base} is the inference time of a single sub-model, and $T_{ensemble}$ is the time overhead of the ensemble strategy (e.g., weighted averaging), which is generally negligible.

Thus, although the inference time of SUDG is proportional to n, the absolute magnitude of this time is the key factor determining deployment feasibility. If T_{infer_base} is sufficiently small, the total inference time can still meet stringent real-time requirements even when n reaches a considerable value.

2. Experimental setup

To accurately quantify the theoretical analysis above, we conducted experiments in a controlled and transparent hardware and software environment. All experiments were performed on a server equipped with dual NVIDIA Tesla T4 graphics processing units (GPUs) (each with 16 GB of memory). The T4 GPU is a mainstream computational card widely adopted in the industry for cloud inference and medium-load training, making its performance metrics highly representative.

The software configuration included PyTorch 1.12.1 and CUDA 11.3. To ensure precise measurements of memory usage and execution time, we utilized PyTorch's built-in memory analysis tools and high-precision timers.

To accelerate the training process, we leveraged both T4 GPUs to train the n sub-models in parallel. This setup simulates real-world research scenarios where available computational resources are used to rapidly iterate model development. All inference speed tests were conducted on a single T4 GPU to emulate the most common deployment setting and to ensure fair and comparable evaluation results.

3. Experimental results

We measured key performance metrics at different ensemble sizes ($n = 3,4,5$) and compared them against a traditional CNN-LSTM baseline model. The results are summarized in the Table VII.

Figure 10, the plot of training time versus n, clearly demonstrates a linear growth trend in total training time with respect to the number of models. This relationship is supported by an exceptionally high goodness of fit ($R^2 = 0.999951$) for the linear regression.

Based on the experimental results presented above, we analyze the feasibility of real-time deployment for SUDG from the perspectives of both training cost and inference efficiency.

TABLE VII. Comparative analysis of SUDG computational efficiency.

Number of sub-models (n)	Total training time (seconds)	Inference time/samples (milliseconds)	Peak memory (MB)
3	48.60	0.2995	59.86
4	64.98	0.2901	68.87
5	81.76	0.3285	78.56
Baseline (CNNLSTM)	10.47	1.9288	42.51

(1) Training cost

The experimental results indicate that the absolute time required for SUDG training is exceptionally low, with all durations remaining within the minute range. Even for the largest model configuration ($n = 5$), the total training time is only 81.76 seconds. This magnitude of time cost represents a significant advantage within academic research or model development cycles. The low training cost enables frequent model updates within short time frames, demonstrating high training efficiency.

The linear relationship between training time and n further implies that researchers can flexibly adjust the ensemble scale based on accuracy requirements and make precise projections of the necessary computational resources. This characteristic underscores the method's favorable scalability and controllability.

Although the baseline CNN-LSTM model exhibits a shorter training time (10.47 seconds), its ability to recognize different classes is significantly less balanced compared to SUDG in small-sample scenarios, particularly due to imbalanced data distribution, resulting in weaker performance on tail classes. Consequently, the marginally longer training time of SUDG can be viewed as an efficient investment toward obtaining a more balanced and reliable model.

(2) Inference efficiency

SUDG demonstrates inference latencies consistently around 0.3 ms. This indicates that SUDG is capable of supporting tasks requiring instant control, security alerts, or highly natural interactions. Furthermore, SUDG achieves a high throughput exceeding 3000 samples per second, imposing a low computational load on the GPU and thereby allocating ample time budget for preprocessing and other pipeline stages.

The inference speed of SUDG ($n = 5$) is nearly six times faster than that of the baseline CNN-LSTM model. Since the models are lightweight, inference speed does not constitute a limiting factor for the deployment of SUDG. In practical deployment scenarios, the computational resources saved can be allocated to running more complex preprocessing algorithms or to reducing overall system power consumption.

(3) Memory footprint

In the current experimental setup, the maximum memory footprint of SUDG is 78.56 MB, which is on the

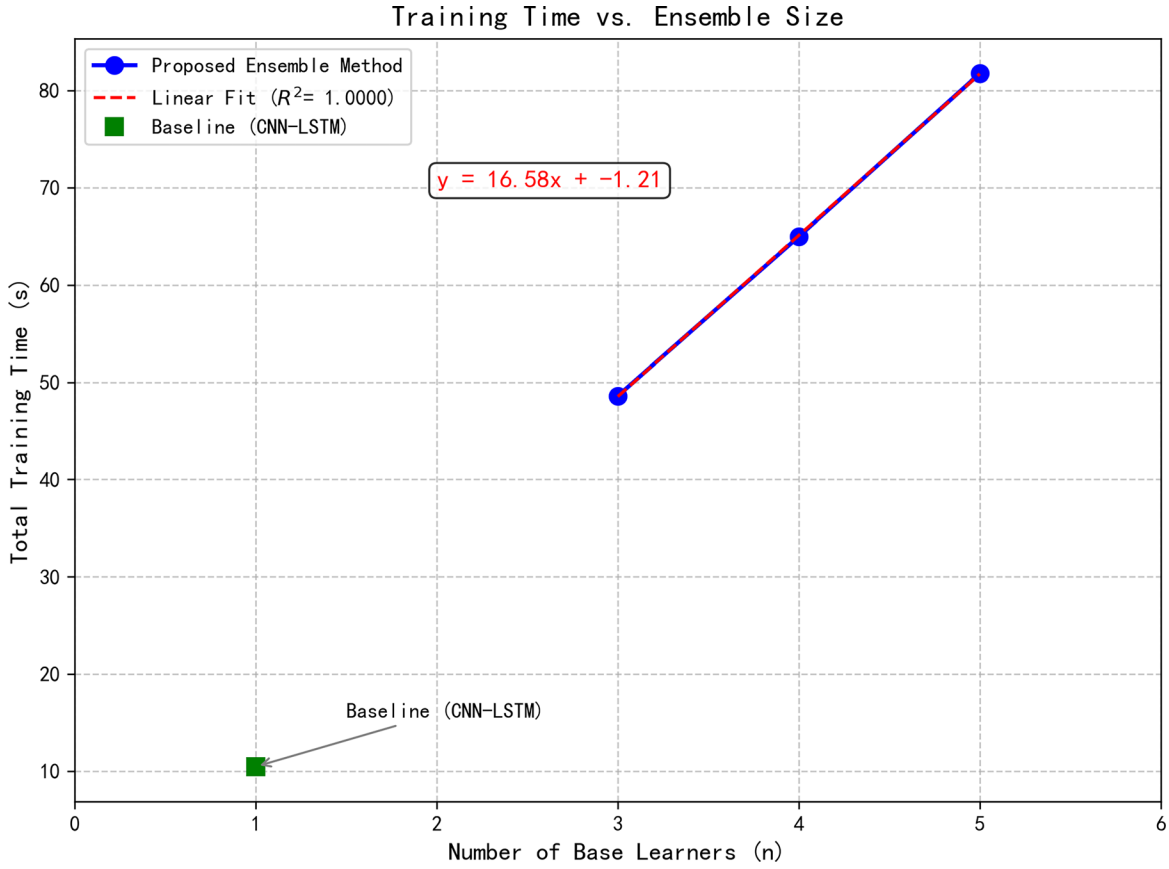

FIG. 10. Training Time as a Function of Ensemble Size.

same order of magnitude as the baseline model (42.51 MB) and remains at a very low level. On a T4 GPU with 16GB of memory, SUDG utilizes less than 0.5% of the available video random-access memory resources. This clearly demonstrates that memory footprint will not become a deployment bottleneck.

(4) Regarding constraints for real-time deployment

Synthesizing the above discussion, we conclude that the computational cost of SUDG does not pose a constraint for its real-time deployment. For the vast majority of real-time auditory tasks, the required processing latency is typically on the order of 200 ms. The sub-millisecond inference latency of SUDG is substantially lower than this requirement.

Even in more demanding scenarios, the forward passes of the n models during inference can be highly parallelized. On hardware with a sufficient number of computational cores, the time required for ensemble inference could approach, or even equal, the inference time of a single model, T_{infer_base}, thereby further reducing latency.

The lightweight nature of SUDG, characterized by low latency and low memory footprint, affords strong compatibility for edge deployment. It is particularly well-suited for deployment on edge computing devices, such as NVIDIA Jetson or Huawei Atlas platforms, where it can operate consistently and stably at high frame rates.

VI. CONCLUSION

This paper presents a deep learning method for UATR under small-sample conditions, which incorporates a data-grouping strategy. The method addresses practical challenges in hydroacoustic recognition, such as insufficient training data due to complex acoustic environments and high acquisition costs, high labeling effort, suboptimal training outcomes, and sensitivity to anomalous samples. The procedure involves partitioning the samples into subgroups for sub-model training, followed by an adjustment of sub-model weights based on prediction outcomes. This design aims to reduce the influence of anomalous samples during training, enhance the learning efficacy of the hydroacoustic recognition model, and improve its final recognition performance.

Before model input, network parameters are initialized. The training set composed of acoustic signals is divided into cross-groups to generate distinct sub-training sets. Corresponding sub-models are then independently trained on these subsets. Sub-models that meet a predefined recognition threshold are initially assigned equal weights; their contributions are then dynamically adjusted in each training iteration according to their recognition accuracy and loss

values. Consequently, sub-models with lower accuracy and higher loss are assigned reduced weights, whereas better-performing sub-models receive higher weights. After training, the final output of the global model is obtained from the weighted aggregation of all sub-model outputs. This strategy aims to improve the utilization of training samples while mitigating the impact of anomalous samples on the model. To further optimize learning under small-sample constraints, additional analyses are conducted regarding dataset allocation ratios, class balance, and the number of groups.

Experiments performed on an acoustic dataset of large yellow croakers showed that the three models using data grouping exhibited noticeable performance gains compared to baseline models without grouping. This validates the practical utility of the proposed approach within this specific application scenario, providing a feasible basis for its potential use in practical engineering contexts.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62271423; in part by Basic Research Program of Science and Technology of Shenzhen, China, under Grant No. JCYJ20230807091406013; in part by State Key Laboratory of Acoustics and Marine Information, Chinese Academy of Sciences under Grant No. SKLA202505; and in part by Fujian Ocean Innovation Center under Grant No. 25FV0CZJ03.

The authors would like to thank Jianming Wu, Shenqin Huang, Dr. Xiaokang Zhang, and Ying Su from Xiamen University for their contributions to the discussion of this research.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

DATA AVAILABILITY

The self-collected *Larimichthys crocea* (large yellow croaker) dataset is not publicly available due to ongoing research but is available from the corresponding author upon reasonable request.

APPENDIX: μ SETTING

To verify the robustness of the method in selecting key parameters, we conducted a parameter sensitivity analysis experiment for the μ setting.

On the small-sample dataset of yellow croaker, we systematically evaluated the impact of two parameters, the marginal improvement value δ (range: 0.01 to 0.03) and the compliance ratio (range: 60% to 90%), on the final model performance. All other settings were consistent with the experimental setup in the main text.

TABLE VIII. Sensitivity analysis of threshold parameters δ and stability ratio.

δ value	Stability ratio	Number of sub-model	Accuracy (%)	F1-score (%)
0.01	60%	5	89.4	86.4
0.01	70%	4	90.4	85.5
0.01	80%	4	88.3	83.9
0.01	90%	1	92.3	89.9
0.02	60%	5	88.1	83.0
0.02	70%	5	89.4	86.7
0.02	80%	4	88.0	82.8
0.02	90%	0	/	/
0.03	60%	5	89.4	87.1
0.03	70%	4	87.2	82.7
0.03	80%	4	88.3	83.2
0.03	90%	0	/	/

The experimental results are shown in the Table VIII. When δ is within the range of 0.01, 0.03 and the compliance ratio is within the range of 60%, 80%, the final performance fluctuation of the combined model is within the fluctuation range of the method performance. This confirms that the method has good robustness within this parameter range. When the compliance ratio is above 90%, due to the low passing rate of the sub-models, no or only one sub-model is selected into the overall model. Although there are a few cases with better performance, the robustness is too low.

[1]Y. L. Zhou, X. M. Xu, X. H. Zhang, L. F. Huang, F. G. Xiao, and Y. W. He, "Vocalization behavior differs across reproductive stages in cultured large yellow croaker *Larimichthys crocea* (Perciformes: Sciaenidae)," Aquaculture **556**, 738267 (2022).
[2]L. T. Yan, Y. Jiang, Q. Xu, G. M. Ding, X. Y. Chen, and M. Liu, "Reproductive dynamics of the large yellow croaker *Larimichthys crocea* (Sciaenidae), a commercially important fishery species in China," Front. Mar. Sci. **9**, 868580 (2022).
[3]Y. Bai, J. Wang, J. Zhao, Q. Ke, A. Qu, Y. Deng, and P. Xu, "Genomic selection for visceral white-nodules diseases resistance in large yellow croaker," Aquaculture **559**, 738421 (2022).
[4]X. M. Ren, D. Z. Gao, Y. L. Yao, F. Yang, J. F. Liu, and F. J. Xie, "Occurrence and characteristics of sound in large yellow croaker (*Pseudosciaena crocea*)," J. Dalian Ocean Univ. **22**(2), 123–128 (2007).
[5]H. Yang, K. Lee, Y. Choo, and K. Kim, "Underwater acoustic research trends with machine learning: General background," J. Ocean Eng. Technol. **34**(2), 147–154 (2020).
[6]D. Li, F. Liu, T. Shen, L. Chen, and D. Zhao, "Data augmentation method for underwater acoustic target recognition based on underwater acoustic channel modeling and transfer learning," Appl. Acoust. **208**, 109344 (2023).
[7]S. Yang, Y. Lu, Y. Wei, J. Zhu, X. Tu, Y. Yang, and F. Qu, "A feasibility study of cross-medium direct acoustic communication between underwater and airborne nodes," J. Mar. Sci. Eng. **12**(12), 2340 (2024).
[8]S. D. Richards, A. D. Heathershaw, and P. D. Thorne, "The effect of suspended particulate matter on sound attenuation in seawater," J. Acoust. Soc. Am. **100**(3), 1447–1450 (1996).
[9]T. C. Bailey, T. Sapatinas, K. J. Powell, and W. J. Krzanowski, "Signal detection in underwater sound using wavelets," J. Am. Stat. Assoc. **93**(441), 73–83 (1998).
[10]S. Xu, M. Ge, J. Feng, X. Wei, H. Tan, Z. Liang, and G. Tong, "Epidemiological investigation on diseases of *Larimichthys crocea* in Ningbo culture area," Front. Cell. Infect. Microbiol. **14**, 1420995 (2024).
[11]J. W. Pridgeon and P. H. Klesius, "Major bacterial diseases in aquaculture and their vaccine development," CABI Rev. **2012**, 1–16.
[12]J. Shu, Z. Xu, and D. Meng, "Small sample learning in big data era," arXiv:abs/1808.04572 (2018).

[13]P. A. Van Walree and R. Otnes, "Ultrawideband underwater acoustic communication channels," IEEE J. Ocean. Eng. **38**(4), 678–688 (2013).
[14]A. Preciado-Grijalva, B. Wehbe, M. B. Firvida, and M. Valdenegro-Toro, "Self-supervised learning for sonar image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, New Orleans, LA (June 2022) (IEEE, New York, 2022), pp. 1499–1508.
[15]T. T. Chungath, A. M. Nambiar, and A. Mittal, "Transfer learning and few-shot learning based deep neural network models for underwater sonar image classification with a few samples," IEEE J. Oceanic Eng. **49**(1), 294–310 (2024).
[16]Y. Chen, Q. Ma, J. Yu, and T. Chen, "Underwater acoustic object discrimination for few-shot learning," in *2019 4th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, Hohhot, China (October 24–26, 2019) (IEEE, New York, 2019), pp. 430–434.
[17]Z. Pu, Q. Zhang, Y. Xue, P. Zhu, and X. Cui, "A novel multi-feature fusion model based on pre-trained wav2vec 2.0 for underwater acoustic target recognition," Remote Sens. **16**(13), 2442 (2024).
[18]X. Yang, H. Yu, H. Sheng, W. Zeng, Q. He, and J. Tu, "An underwater acoustic target recognition method based on transfer learning," in *2024 9th International Conference on Electronic Technology and Information Science (ICETIS)*, Hangzhou, China (May 17–19, 2024) (IEEE, New York, 2024), pp. 506–510.
[19]X. Wang, P. Wu, B. Li, G. Zhan, J. Liu, and Z. Liu, "A self-supervised dual-channel self-attention acoustic encoder for underwater acoustic target recognition," Ocean Eng. **299**, 117305 (2024).
[20]W. Qiang, Z. Song, Z. Gu, J. Li, C. Zheng, F. Sun, and H. Xiong, "On the generalization and causal explanation in self-supervised learning," Int. J. Comput. Vision **133**, 1727–1754 (2025).
[21]D. Liu, I. W. Tsang, and G. Yang, "A convergence path to deep learning on noisy labels," IEEE Trans. Neural Netw. Learn. Syst. **35**(4), 5170–5182 (2024).
[22]P. C. Etter, *Underwater Acoustic Modeling and Simulation* (CRC, Boca Raton, FL, 2018).
[23]L. Zhang, D. Wu, X. Han, and Z. Zhu, "Feature extraction of underwater target signal using Mel frequency cepstrum coefficients based on acoustic vector sensor," J. Sens. **2016**(1), 7864213 (2016).
[24]W. Wang, S. Li, J. Yang, Z. Liu, and W. Zhou, "Feature extraction of underwater target in auditory sensation area based on MFCC," in *2016 IEEE/OES China Ocean Acoustics (COA)*, Harbin, China (January 9–11, 2016) (IEEE, New York, 2016), pp. 1–6.
[25]S. K. Kopparapu and M. Laxminarayana, "Choice of Mel filter bank in computing MFCC of a resampled speech," in *10th International Conference on Information Science, Signal Processing and Their Applications* (ISSPA), Kuala Lumpur (May 2010) (IEEE, New York, 2010), pp. 121–124.
[26]D. F. N. Kong, C. Shen, C. Tian, and S. R. Zhang, "Underwater acoustic monitoring: A comprehensive approach to enhance MFCC robustness and classification accuracy" (2024).
[27]G. M. Wenz, "Review of underwater acoustics research: Noise," J. Acoust. Soc. Am. **51**(3B), 1010–1024 (1972).
[28]X. Luo, M. Zhang, T. Liu, M. Huang, and X. Xu, "An underwater acoustic target recognition method based on spectrograms with different resolutions," J. Mar. Sci. Eng. **9**(11), 1246 (2021).
[29]Y. Dong, X. Shen, Z. Jiang, and H. Wang, "Recognition of imbalanced underwater acoustic datasets with exponentially weighted cross-entropy loss," Appl. Acoust. **174**, 107740 (2021).
[30]B. Vrigazova, "The proportion for splitting data into training and test set for the bootstrap in classification problems," Bus. Syst. Res. J. **12**(1), 228–242 (2021).
[31]J. Salamon, C. Jacoby, and J. P. Bello, "A dataset and taxonomy for urban sound research," in *Proceedings of the 22nd ACM International Conference on Multimedia*, Orlando, FL (November 2014) (Association for Computing Machinery, New York, 2014), pp. 1041–1044.
[32]M. Avadhani and A. P. Bidargaddi, "Multi-class urban sound classification with deep learning architectures," in *2024 5th International Conference for Emerging Technology (INCET)*, Belgaum, India (May 24–26, 2024) (IEEE, New York, 2024), pp. 1–7.
[33]P. Malaviya, Y. Kumar, and N. Modi, "Advancements in environmental sound classification: Evaluating machine learning and deep learning approaches on the UrbanSound8k," in *2023 Seventh International Conference on Image Information Processing (ICIIP)*, Solan, India (November 2023) (IEEE, New York, 2023), pp. 900–905.
[34]K. Z. Mon, K. Galajit, C. O. Mawalim, J. Karnjana, T. Isshiki, and P. Aimmanee, "Spoof detection using voice contribution on LFCC features and ResNet-34," in *2023 18th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, Bangkok, Thailand (November 27–29, 2023) (IEEE, New York, 2023), pp. 1–6.
[35]R. Zaheer, I. Ahmad, D. Habibi, K. Y. Islam, and Q. V. Phung, "A survey on artificial intelligence-based acoustic source identification," IEEE Access **11**, 60078–60108 (2023).
[36]M. Xu, Z. Yin, M. Wu, Z. Wu, Y. Zhao, and Z. Gao, "Spectrum sensing based on parallel CNN–LSTM network," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium (May 2020) (IEEE, New York, 2020), pp. 1–5.
[37]S. H. Bae, I. K. Choi, and N. S. Kim, "Acoustic scene classification using parallel combination of LSTM and CNN," in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2016 Workshop (DCASE2016)*, Budapest, Hungary, 2016, pp. 11–15.
[38]M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT (June 2018) (IEEE, New York, 2018), pp. 4510–4520.
[39]J. Chen, F. Zhang, and Y. Li, "A sound event recognition method of crop shear scrap falling state based on Log-Mel spectrogram and MobileNetV2," in *2023 42nd Chinese Control Conference (CCC)*, Tianjin, China (July 24–26, 2023) (IEEE, New York, 2023), pp. 6946–6951.
[40]G. Ramesh, C. T. Puttaraj, K. S. Sashreeth, P. Naidu, P. B. Ashish, and H. Kunder, "Music genre classification using CNN and MobileNetV2," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, Kamand, India (June 24–28, 2024) (IEEE, New York, 2024), pp. 1–7.
[41]X. Liao, Y. Wu, N. Jiang, J. Sun, W. Xu, S. Gao, and Q. Li, "Automated detection of abnormal respiratory sound from electronic stethoscope and mobile phone using MobileNetV2," Biocyber. Biomed. Eng. **43**(4), 763–775 (2023).