

Article

Learnable Two-Stage Frequency-Guided Front-End for Robust Underwater Acoustic Target Recognition

Heng Zhang ^{1,2}, Haiyan Wang ^{1,*}, Shaodong Zhang ³ and Yongsheng Yan ¹¹ School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China² Yichang Testing Technique R&D Institute, Yichang 443003, China³ Marine Design and Research Institute of China, Shanghai 200011, China

* Correspondence: hywang@sust.edu.cn

Abstract

Underwater acoustic target recognition is a key enabling technology for marine environmental monitoring, resource exploration, and maritime defense. However, ship-radiated noise is strongly affected by nonstationary propagation, multipath interference, frequency-dependent attenuation, and time-varying ambient noise, making robust acoustic representation a central challenge. Most existing deep learning-based recognition methods still rely on fixed time–frequency front-ends, such as Mel-spectrograms or wavelet scattering representations, whose analysis parameters are predefined and cannot be adapted to the recognition objective. This separation between acoustic preprocessing and deep model optimization may weaken narrowband tonal components, modulation-related structures, or weak target-dependent frequency patterns before they are processed by the classifier. To address this limitation, this paper proposes a Learnable Two-Stage Frequency-Guided Front-End, termed LTFF, for underwater acoustic signal recognition. In the first stage, LTFF performs frequency-guided discrimination by decomposing the input waveform into frequency-dependent components and applying learnable complex Gabor filters, whose center frequencies and bandwidths are optimized through end-to-end training. In the second stage, squared-modulus envelope extraction, learnable Gaussian pooling, and per-channel energy normalization are integrated to improve local temporal stability and attenuate slowly varying energy fluctuations. In this way, LTFF establishes a trainable connection between physically meaningful underwater acoustic structures and downstream deep recognition models. Experiments on two public underwater acoustic datasets with convolutional and Transformer-based backbones show that LTFF provides a task-adaptive alternative to fixed acoustic front-ends and is particularly compatible with convolutional architectures. These results indicate that learnable frequency-guided front-ends are promising for improving robust underwater acoustic recognition under dynamic marine conditions.

Keywords: learnable acoustic front-end; complex Gabor filter bank; frequency-guided front-end; underwater acoustic target recognition; per-channel energy normalization



Academic Editor: Weicheng Cui

Received: 21 May 2026

Revised: 23 June 2026

Accepted: 24 June 2026

Published: 30 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Underwater acoustic target recognition (UATR) is a key enabling technology for marine environmental monitoring, resource exploration, and maritime defense. However, the underwater acoustic channel is substantially more complex and variable than many terrestrial acoustic scenarios. The received ship-radiated signals are jointly affected by sea-surface fluctuation, seabed reflection, water-column stratification, platform motion,

and region-dependent ambient interference. These factors lead to attenuation, scattering, reverberation, multipath propagation, Doppler distortion, and time-varying signal-to-noise ratios (SNRs). As a result, underwater acoustic signals often exhibit pronounced nonstationary, nonlinear, and non-Gaussian characteristics. These properties make fixed handcrafted preprocessing or static time–frequency representations insufficient for robust recognition, because discriminative frequency components may be weakened or distorted before they are delivered to the recognition model. Therefore, an effective UATR system should not only learn target-discriminative acoustic signatures, but also learn an adaptive front-end representation that is adapted to the dynamic propagation characteristics of underwater acoustic signals [1,2].

Traditional UATR systems typically consist of acoustic feature extraction and subsequent recognition using statistical or deep models. Representative handcrafted features include STFT-based time–frequency spectra, Mel-frequency representations, GFCCs, LO-FAR/DEMON spectra, wavelet-based descriptors, and other task-specific acoustic representations. With the development of deep learning, recent studies have further improved recognition performance by learning hierarchical representations from spectrograms or raw waveforms [3–9]. Despite these advances, a fundamental limitation remains in many existing pipelines: the acoustic front-end is usually fixed before model training. Parameters such as window length, filter shape, center frequency, bandwidth, and smoothing scale are commonly determined by empirical rules or task-independent acoustic priors. This separation between signal preprocessing and recognizer optimization can limit adaptation to underwater acoustic variability for underwater acoustic recognition, because frequency-dependent attenuation, multipath propagation, Doppler distortion, and nonstationary interference can alter the discriminative structure of ship-radiated noise. As a result, narrowband mechanical line spectra, modulation-related components, or weak target-dependent frequency patterns may be smoothed, distorted, or suppressed before reaching the trainable recognizer.

Learnable acoustic front-ends provide a promising strategy for reducing this mismatch between static preprocessing and data-driven recognition. By parameterizing the filter bank and optimizing its parameters jointly with the downstream recognizer, the front-end can learn task-relevant frequency responses directly from underwater acoustic data [10–14]. However, designing a learnable front-end for UATR is not equivalent to simply replacing fixed filters with trainable filters. An effective underwater acoustic front-end should satisfy three requirements. First, it should preserve frequency-selective structures that are physically meaningful for ship-radiated noise, such as tonal components, broadband energy distributions, and modulation-related patterns. Second, it should remain stable under small local temporal shifts and distortions caused by multipath propagation, platform motion, and time-varying propagation delays. Third, it should suppress slowly varying background interference and amplitude fluctuations without eliminating weak but discriminative target-related components. These requirements indicate that a learnable UATR front-end should combine interpretable signal-processing operations, acoustic robustness, and end-to-end trainability.

Motivated by these considerations, this paper proposes a Learnable Two-Stage Frequency-Guided Front-End method, termed LTFF, for underwater acoustic signal recognition. LTFF is designed as a differentiable acoustic front-end that connects underwater signal characteristics with deep recognition models. In the first stage, LTFF performs frequency-guided discrimination by combining coarse frequency-band partitioning with learnable complex Gabor filtering. This design enables the front-end to capture band-dependent acoustic variations and emphasize frequency regions that are more informative for target recognition. In the second stage, LTFF performs robustness enhancement through

squared-modulus envelope extraction, learnable Gaussian pooling, and per-channel energy normalization. The squared-modulus operation converts complex filter responses into local energy envelopes, Gaussian pooling improves local time-shift stability, and PCEN provides adaptive gain control for suppressing background fluctuations. Through these two stages, LTFF transforms acoustic preprocessing from a fixed handcrafted operation into a parameterizable module that can be optimized jointly with the recognizer. Furthermore, LTFF is evaluated using both ResNet18 and Vision Transformer backbones to investigate how different architectural inductive biases interact with the proposed learnable acoustic representation.

The main contributions of this paper are summarized as follows.

A learnable two-stage acoustic front-end is proposed for underwater acoustic target recognition. We develop a Learnable Two-Stage Frequency-Guided Front-End, termed LTFF, to replace fixed handcrafted acoustic preprocessing with a trainable representation module. The proposed front-end integrates frequency-band partitioning, learnable complex Gabor filtering, squared-modulus envelope extraction, learnable Gaussian pooling, and per-channel energy normalization into a unified pipeline. In this way, the acoustic representation can be optimized according to the recognition objective rather than being determined by manually selected time–frequency parameters.

A frequency-guided feature learning mechanism is established to adapt to the spectral characteristics of ship-radiated noise. Unlike conventional fixed front-ends that process the spectrum using predefined windows or filter banks, LTFF first decomposes the input signal into coarse frequency bands and then applies learnable Gabor filters within each band. This design allows the model to adaptively emphasize target-related tonal components, modulation structures, and broadband energy patterns while reducing the influence of less informative or noise-dominated frequency regions. Therefore, LTFF provides a more task-adaptive acoustic representation for dynamic underwater environments.

A theoretically consistent formulation of LTFF is provided from the perspective of differentiable signal processing. We formulate each operation in LTFF using discrete signal-processing expressions, including frequency-domain masking, complex Gabor convolution, squared-modulus energy extraction, Gaussian pooling, and PCEN-based adaptive normalization. This formulation clarifies the trainable parameters, tensor flow, and optimization path of the proposed front-end, showing that LTFF is differentiable or almost everywhere differentiable and can be jointly optimized with downstream recognition networks.

2. Method and Theory

2.1. Learnable-Parameter Two-Stage Frequency-Guided Feature

Underwater acoustic signals contain frequency-dependent target information arising from mechanical vibration, propeller modulation, broadband hydrodynamic radiation, and propagation-induced spectral distortion. Conventional fixed front-ends usually adopt predefined windows, filters, or frequency scales, which may be insufficient to preserve target-discriminative components under dynamic marine environments. To address this issue, we propose a Learnable Two-Stage Frequency-Guided Front-End method, termed LTFF, which formulates underwater acoustic preprocessing as a differentiable and trainable front-end. As illustrated in Figure 1, LTFF consists of two sequential stages: frequency-guided discrimination and robustness enhancement. The first stage extracts band-dependent acoustic responses using frequency partitioning and learnable complex Gabor filtering. The second stage converts complex responses into stable energy representations and improves robustness through Gaussian pooling and per-channel energy normalization.

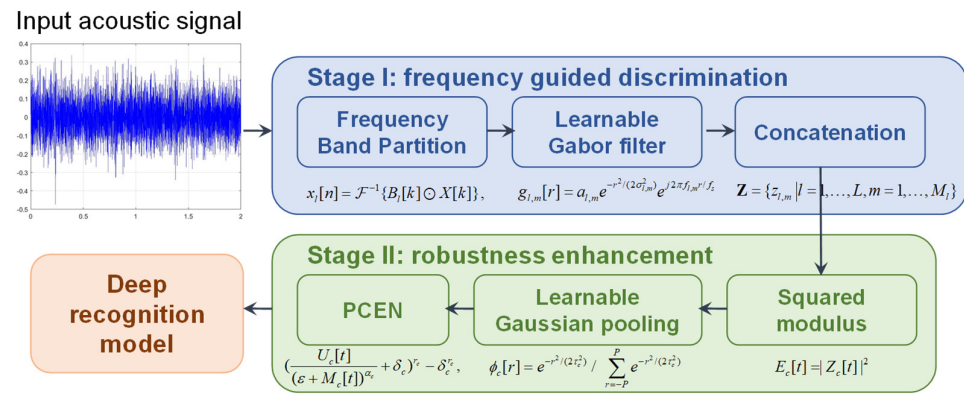


Figure 1. LTFF architecture.

Let $x[n] \in \mathbb{R}^T$ denote an input underwater acoustic signal with length T and sampling frequency f_s . LTFF maps $x[n]$ into a two-dimensional feature representation

$$F_{\text{LTFF}} = \Phi_{\Theta}(x) \in \mathbb{R}^{C \times T'}, \quad (1)$$

where C denotes the number of output channels, T' is the temporal length after pooling or downsampling, $\Phi_{\Theta}(\cdot)$ denotes the LTFF front-end, and Θ contains all learnable parameters, including Gabor center frequencies, Gabor bandwidths, Gaussian pooling scales, and PCEN parameters.

To clarify the novelty of our approach, we summarize the differences between LTFF and representative prior methods, including LEAF [10], UALF [13], learnable wavelet-based methods, and conventional fixed front-ends such as Mel filters and wavelet filters, as depicted in Table 1. Unlike previous methods, LTFF explicitly partitions the input signal into low-, mid-, and high-frequency bands before applying learnable Gabor filters. This frequency-guided decomposition allows different filter groups to specialize in physically meaningful acoustic regions related to ship-radiated noise. Combined with squared-modulus energy extraction, learnable Gaussian pooling, and per-channel energy normalization, LTFF provides enhanced local stability and background suppression.

Table 1. Comparison Table. Filter Type: Indicates whether the front-end uses fixed or trainable filters. Frequency Constraint: Whether filter learning is constrained to specific frequency bands. Underwater Prior: Whether the method considers underwater acoustic characteristics, e.g., ship-radiated frequency patterns. Adaptive Normalization: Whether a learnable normalization (e.g., PCEN) is applied to improve robustness against energy variation and background noise.

Method	Filter Type	Frequency Constraint	Underwater Prior	Adaptive Normalization
Mel	Fixed filter bank	No	No	No
Wavelet	Fixed wavelet filters	Yes	No	No
LEAF	Learnable Gabor filters	Yes	No	PCEN
UALF	Learnable filters	No	Yes	-
Learnable wavelet-based methods	Learnable wavelet filters	-	No	No
LTFF (Ours)	Learnable Gabor filters	Yes	Yes	PCEN

2.1.1. Frequency-Band Partitioning

To explicitly model the frequency-dependent structure of ship-radiated noise, the input signal is first decomposed into L non-overlapping frequency bands. In this study,

$L = 3$, corresponding to low-, mid-, and high-frequency components. The discrete Fourier transform of $x[n]$ is denoted as

$$X[k] = \mathcal{F}\{x[n]\}, k = 0, 1, \dots, T - 1. \quad (2)$$

For the l -th frequency band, a binary band-selection mask $B_l[k]$ is defined as

$$\sum_{l=1}^L B_l[k] = 1, B_i[k]B_j[k] = 0, i \neq j, \quad (3)$$

and a smooth band-selection mask $B_l[k]$ is

$$\sum_{l=1}^L B_l[k] = 1, 0 \leq B_l[k] \leq 1. \quad (4)$$

The original signal $x[n]$ is real. If we want $x_l[n]$ to remain a real signal, the frequency-domain mask must be conjugate symmetry $B_l[k] = B_l[(T - k) \bmod T]$. The band-limited signal is obtained by

$$x_l[n] = \mathcal{F}^{-1}\{B_l[k] \odot X[k]\}, \quad (5)$$

where $\odot$ denotes element-wise multiplication. This operation ensures that subsequent learnable filters operate on acoustically meaningful frequency regions rather than the full spectrum indiscriminately.

2.1.2. Learnable Complex Gabor Filtering

Within each frequency band, LTFF applies a bank of learnable complex Gabor filters to capture local time–frequency structures. For the l -th band and the m -th filter, the discrete Gabor kernel is formulated as

$$g_{l,m}[r] = a_{l,m} \exp(-r^2 / (2\sigma_{l,m}^2)) \exp(j2\pi f_{l,m} r / f_s), r = -R, \dots, R, \quad (6)$$

where $a_{l,m}$ is an amplitude factor, $f_{l,m}$ is the learnable center frequency, $\sigma_{l,m}$ controls the temporal spread of the Gabor kernel and is inversely related to its frequency bandwidth, and $2R + 1$ is the filter length. To ensure physically meaningful filters during training, the parameters are constrained as

$$f_{l,m} \in [f_l^{\min}, f_l^{\max}], \sigma_{l,m} > 0. \quad (7)$$

In practice, these constraints can be implemented by reparameterizing $f_{l,m}$ with a sigmoid function and $\sigma_{l,m}$ with a softplus function. The complex filter response is then computed by one-dimensional convolution:

$$z_{l,m}[t] = (x_l * g_{l,m})[t]. \quad (8)$$

The responses from all frequency bands and filters are concatenated along the channel dimension:

$$\mathbf{Z} = \text{Concat}(\{z_{l,m} | l = 1, \dots, L, m = 1, \dots, M_l\}) \in \mathbb{C}^{C \times T}, C = \sum_{l=1}^L M_l, \quad (9)$$

where M_l denotes the number of filters in the l -th band. This representation preserves complex-valued local spectral responses before energy conversion.

2.1.3. Squared-Modulus Energy Extraction

Since the Gabor responses are complex-valued, LTFF converts them into real-valued local energy envelopes using the squared-modulus operation:

$$E_c[t] = |Z_c[t]|^2 = \text{Re}(Z_c[t])^2 + \text{Im}(Z_c[t])^2, c = 1, \dots, C. \quad (10)$$

This operation reduces sensitivity to local phase variations while retaining the energy distribution of each sub-band response. For underwater acoustic signals, such an energy-envelope representation is suitable for describing target-related spectral patterns that may be affected by propagation delay, phase distortion, and multipath interference.

2.1.4. Learnable Gaussian Pooling

Although the squared-modulus operation provides local energy envelopes, the resulting responses may still be sensitive to small temporal shifts. To improve local temporal stability, LTFF applies learnable Gaussian pooling to each channel:

$$\phi_c[r] = \exp(-r^2/(2\tau_c^2)) / \sum_{r=-P}^P \exp(-r^2/(2\tau_c^2)), r = -P, \dots, P, \quad (11)$$

where $\tau_c > 0$ is a learnable pooling scale and $2P + 1$ is the pooling window length. The smoothed energy response is obtained as

$$U_c[t] = (E_c * \tau_c)[t]. \quad (12)$$

Compared with fixed average pooling, learnable Gaussian pooling allows the front-end to adapt the degree of local smoothing to different frequency channels. A smaller τ_c preserves rapid local variations, whereas a larger τ_c enhances stability against temporal fluctuations.

2.1.5. Per-Channel Energy Normalization

Underwater acoustic recordings are often contaminated by slowly varying background noise and amplitude fluctuations. To suppress such interference while preserving weak target-related components, LTFF further applies per-channel energy normalization (PCEN). For each channel, an exponential moving average is first computed:

$$M_c[t] = (1 - s_c)M_c[t - 1] + s_c U_c[t], 0 < s_c < 1. \quad (13)$$

where $s_c \in (0, 1)$ is a learnable smoothing coefficient. The normalized feature is then defined as

$$F_c[t] = (U_c[t] / (\varepsilon + M_c[t])^{\alpha_c} + \delta_c)^{r_c} - \delta_c^{r_c}, \quad (14)$$

where $\varepsilon > 0$ is a small constant for numerical stability, and $0 \leq \alpha_c \leq 1$, $\delta_c > 0$, and $r_c > 0$ are learnable parameters controlling gain normalization, bias compensation, and dynamic-range compression, respectively. PCEN can be interpreted as a learnable adaptive gain-control mechanism. Unlike static logarithmic compression, it adjusts the normalization strength according to channel-wise energy dynamics, making the representation more robust to slowly varying noise and amplitude scaling. PCEN can reduce sensitivity to multiplicative amplitude scaling, especially when α_c approaches 1 and the moving average tracks the slow energy variation.

The final LTFF representation is

$$F_{\text{LTFF}} = [F_1, F_2, \dots, F_C] \in \mathbb{R}^{C \times T'}. \quad (15)$$

This feature map can be directly fed into a downstream recognition backbone. Since frequency partitioning, Gabor convolution, squared-modulus transformation, Gaussian pooling, and PCEN are differentiable or almost everywhere differentiable, LTFF can be trained jointly with the recognizer under the recognition loss. Given a recognizer $h_{\Psi}(\cdot)$ with parameters Ψ , the overall optimization objective is

$$\min_{\Theta, \Psi} (1/N) \sum_{i=1}^N \mathcal{L}_{CE}(h_{\Psi}(\Phi_{\Theta}(x_i)), y_i), \quad (16)$$

where $\mathcal{L}_{CE}$ denotes the cross-entropy loss, x_i is the input signal, and y_i is its class label. Therefore, LTFF transforms underwater acoustic preprocessing from a fixed handcrafted procedure into a learnable front-end that is optimized according to the recognition objective. The process of LTFF is as shown in the Algorithm 1.

Algorithm 1 The process of LTFF

Input: Acoustic signal $x[n] \in \mathbb{R}^T$, sampling frequency f_s , band masks $\{B_l\}_{l=1}^L$.
Output: LTFF feature map $\mathbf{F}_{\text{LTFF}} \in \mathbb{R}^{C \times T'}$.
 1: Compute the frequency spectrum $X[k] = \mathcal{F}\{x[n]\}$.
 2: Decompose $x[n]$ into L frequency-band signals: $x_l[n] = \mathcal{F}^{-1}\{B_l[k] \odot X[k]\}$.
 3: For each band $x_l[n]$, apply learnable complex Gabor filters $g_{l,m}$ to obtain $z_{l,m}[t] = (x_l * g_{l,m})[t]$.
 4: Concatenate all filter responses: $\mathbf{Z} = \text{Concat}(\{z_{l,m}\})$.
 5: Convert complex responses into energy envelopes: $E_c[t] = \text{Re}^2(\mathbf{Z}_c[t]) + \text{Im}^2(\mathbf{Z}_c[t])$.
 6: Apply learnable Gaussian pooling: $U_c[t] = (E_c * \phi_c)[t]$.
 7: Estimate channel-wise background energy by exponential moving average: $M_c[t] = (1 - s_c)M_c[t-1] + s_c U_c[t]$.
 8: Perform PCEN normalization to obtain $F_c[t]$.
 9: Construct the final feature map: $\mathbf{F}_{\text{LTFF}} = [F_1, F_2, \dots, F_C]$.
 10: Optimize LTFF parameters jointly with the recognizer using cross-entropy loss.

2.2. Theoretical Analysis of LTFF

This subsection provides a theoretical interpretation of the proposed LTFF front-end and clarifies why the operations introduced in Section 2.1 are suitable for underwater acoustic signal recognition. The analysis follows the same order as the method pipeline: frequency-band decomposition, learnable Gabor filtering, squared-modulus envelope extraction, Gaussian pooling, PCEN, and downstream classification. The purpose is not to claim exact physical source separation, but to show that LTFF has well-defined mathematical properties in terms of differentiability, frequency selectivity, local stability, adaptive gain control, and architecture-dependent compatibility.

2.2.1. Differentiability and End-to-End Learnability

Let $F_{\phi}(x)$ denote the LTFF front-end parameterized by ϕ , and let $h_{\theta}(\cdot)$ denote the downstream recognition network. The complete recognition model can be written as

$$p_{\theta}(y|x) = h_{\theta}(F_{\phi}(x)). \quad (17)$$

In LTFF, the FFT-based band partition is a linear transform followed by deterministic frequency masking and inverse transformation. The Gabor filtering stage is a linear convolution with kernels parameterized by learnable center frequencies and bandwidths. The squared-modulus operation, Gaussian smoothing, exponential moving average, and PCEN

are differentiable or almost everywhere differentiable under positive parameter constraints and a nonzero numerical constant epsilon. Therefore, for the training objective

$$L(\phi, \theta) = -(1/Q) \sum_{q=1}^Q \log p_{\theta}(y_q | F_{\phi}(x_q)), \quad (18)$$

The gradient can be propagated from the recognition loss to both the recognizer parameters θ and the acoustic front-end parameters ϕ . This property distinguishes LTFF from fixed front-ends such as handcrafted Mel or wavelet features, whose filter shapes and normalization behavior remain unchanged during model optimization.

The theoretical implication is that LTFF transforms the acoustic front-end from a manually specified preprocessing step into an optimization-coupled representation-learning process. The learned filters are not selected according to a generic perceptual prior, but are updated according to the target-recognition objective. This is particularly important for underwater acoustic signals, whose informative components are determined by target radiation mechanisms and propagation conditions rather than by human auditory perception alone.

2.2.2. Frequency-Selective Representation of Ship-Radiated Noise

Ship-radiated noise generally contains multiple acoustic components, including low-frequency tonal components associated with propulsion and machinery, mid-frequency harmonic structures, broadband cavitation-related components, and environmental noise. A simplified model can be expressed as

$$x[n] = \sum_{r=1}^R a_r[n] \cos(2\pi f_r n / f_s + \varphi_r) + b[n] + \eta[n], \quad (19)$$

where the first term represents tonal or harmonic components, $b[n]$ denotes broadband target-related components, and $\eta[n]$ denotes background noise or interference. Since these components are distributed unevenly over frequency, a single fixed global representation may either smooth out narrowband discriminative lines or retain noise-dominated regions with limited classification value.

LTFF addresses this issue by first dividing the spectrum into low-, middle-, and high-frequency regions and then applying band-specific learnable Gabor filters. For a tonal component near f_r , the magnitude of the Gabor response is maximized when the learnable center frequency $f_{b,n}$ approaches f_r and the bandwidth is sufficiently narrow to isolate the component. Conversely, for broadband structures, a wider bandwidth can preserve distributed energy patterns. Through gradient-based optimization, the center frequencies and bandwidths are encouraged to move toward frequency regions that reduce classification loss. In this sense, LTFF provides a frequency-selective mechanism that can emphasize target-discriminative components while suppressing less informative or noise-dominated bands.

The three-band partition also has a regularizing effect. It restricts each group of filters to a physically meaningful frequency region and prevents all filters from collapsing into the same dominant band. This improves interpretability because the learned responses can be analyzed according to low-, mid-, and high-frequency acoustic evidence. Therefore, the first stage of LTFF is theoretically consistent with the frequency-dependent nature of underwater target radiation and propagation.

2.2.3. Local Stability Induced by Squared Modulus and Gaussian Pooling

The complex Gabor response preserves both amplitude and phase information. However, underwater acoustic recordings are frequently affected by propagation delay, slight window misalignment, and local phase variations. Directly feeding complex responses into a recognizer may increase sensitivity to such nuisance factors. LTFF therefore converts the complex response $y_i[t]$ into a real-valued local energy envelope,

$$E_i[t] = |y_i[t]|^2. \quad (20)$$

This operation removes instantaneous phase ambiguity while retaining the amplitude modulation and envelope patterns that are closely related to mechanical and propulsion signatures.

Gaussian pooling further improves local time-shift stability. Consider a small temporal shift Δ applied to the input signal. Since convolution is shift-equivariant, the corresponding sub-band energy approximately shifts as $E_i^\Delta[t] = E_i[t - \Delta]$. After Gaussian pooling,

$$U_i[t] = (E_i * h_i)[t], \quad (21)$$

where h_i is a normalized Gaussian kernel. For a smooth h_i and a bounded energy response, the change induced by a small shift is bounded by

$$|U_i[t] - U_i^\Delta[t]| \leq |\Delta| \|E_i\|_\infty \|h_i'\|_1. \quad (22)$$

This inequality shows that the pooled response varies continuously with small temporal shifts. The learnable pooling scale s_i controls the trade-off between stability and temporal resolution: a larger s_i yields stronger local stability but may blur rapid transients, whereas a smaller s_i preserves fine temporal details. By learning s_i from data, LTFF adaptively selects an appropriate stability-resolution balance for each frequency channel.

2.2.4. Adaptive Gain Control and Noise Suppression by PCEN

Underwater acoustic signals often exhibit nonstationary background noise and amplitude fluctuations caused by propagation loss, sensor gain variation, or environmental interference. Static logarithmic compression can reduce the dynamic range, but it cannot adapt the degree of normalization to different channels and acoustic conditions. PCEN in LTFF provides a learnable automatic gain-control mechanism. For each channel, the exponential moving average,

$$M_i[t] = (1 - r_i)M_i[t - 1] + r_i U_i[t], \quad (23)$$

estimates the local background energy, and the normalized response is given by

$$Z_i[t] = (U_i[t] / (\epsilon + M_i[t])^{\alpha_i} + \delta_i)^{\gamma_i} - \delta_i^{\gamma_i}. \quad (24)$$

The role of PCEN can be interpreted from scale behavior. If a channel response is multiplied by a slowly varying positive factor c and the moving average tracks the same factor, then $U_i[t]$ and $M_i[t]$ are both scaled by c . The normalized term approximately becomes $c^{1-\alpha_i} U_i[t] / M_i[t]^{\alpha_i}$. When α_i approaches 1, the response becomes nearly invariant to multiplicative amplitude scaling. When α_i is smaller, more absolute energy information is preserved. Therefore, learnable α_i allows LTFF to adaptively determine whether a channel should emphasize relative contrast or absolute energy.

The moving-average mechanism also explains the noise-suppression behavior. Slowly varying background components are gradually absorbed into $M_i[t]$ and are attenuated

after normalization, whereas short-term target-related energy variations are less fully absorbed and can remain salient. The parameters r_i , α_i , δ_i , and γ_i jointly determine the time scale, strength, offset, and compression of this adaptive gain control. Consequently, PCEN improves robustness without imposing a fixed denoising rule that might remove weak but useful target components.

2.2.5. Compatibility with CNN and Transformer Backbones

The proposed LTFF front-end produces localized sub-band energy patterns whose discriminative information is mainly reflected in spectral textures, envelope transitions, and frequency-specific energy distributions. CNNs are naturally compatible with such representations because local connectivity, weight sharing, and hierarchical pooling provide strong inductive biases for extracting local and translation-tolerant patterns [15]. Therefore, a convolutional recognizer can efficiently exploit the structured local patterns generated by LTFF, even when the available underwater acoustic datasets are limited in scale.

In contrast, a vanilla Transformer relies primarily on global self-attention and has weaker built-in locality priors. This makes it powerful for long-range dependency modeling, but it may require larger datasets, stronger pretraining, or more carefully designed patch embeddings to fully exploit a flexible learnable acoustic front-end. Thus, if LTFF yields a larger improvement with CNNs [16] than with ViT [17], the result should not be interpreted as a failure of the front-end. Rather, it reveals an interaction between learnable acoustic representations and backbone inductive biases. This analysis justifies using both ResNet18 and ViT [18] in the experiments: they serve as representative downstream recognizers for evaluating the architecture-dependent behavior of LTFF, rather than as methodological contributions themselves.

2.2.6. Computational Complexity and Parameter Overhead

The additional parameters introduced by LTFF mainly include the center frequencies and bandwidths of the Gabor filters, the Gaussian pooling scales, and the PCEN parameters. If N learnable filters with length W are used for an input of length T , direct convolution has a complexity of $O(NWT)$, while FFT-based implementation can reduce the filtering cost for long signals. Gaussian pooling and PCEN have complexities of $O(NPT)$ and $O(NT)$, respectively, where P is the pooling kernel length. Compared with the downstream CNN or Transformer recognizer, the parameter overhead of LTFF is limited because each channel only introduces a small number of scalar parameters.

Overall, the theoretical analysis demonstrates that LTFF is a differentiable, frequency-selective, locally stable, and adaptively normalized acoustic front-end. These properties directly correspond to the challenges of underwater acoustic signal recognition, including frequency-dependent target signatures, local time shifts, nonstationary noise, and architecture-dependent representation learning. Therefore, LTFF provides a theoretically grounded alternative to a fixed-parameter front-end while preserving compatibility with standard deep learning recognizers.

3. Results

3.1. Datasets

Two public underwater acoustic datasets are used to evaluate the proposed method.

1. ShipsEar [19] contains recordings from 11 types of vessels, with a total duration of approximately 3 h and an original sampling frequency of 52,734 Hz.
2. DeepShip [20] contains more than 47 h of underwater acoustic recordings from 265 ships belonging to four categories. The recordings were collected under dif-

ferent sea states and noise conditions, which makes the dataset suitable for evaluating recognition robustness.

For consistent processing, all audio signals are resampled to 12 kHz. ShipsEar recordings are segmented into 3 s slices, while DeepShip recordings are segmented into 20 s slices unless otherwise specified. To reduce the risk of temporal leakage, we follow the class definitions provided in the original ShipsEar and DeepShip datasets. Before segmentation, recordings shorter than 12 s in ShipsEar and shorter than 80 s in DeepShip are removed. For each retained recording, 75% of its total duration is randomly selected as training data, while the remaining 25% is used as testing data. Fixed-length segmentation is then performed separately on the training and testing portions. This procedure differs from directly randomizing cropped segments after segmentation, because the train/test division is conducted before generating fixed-length samples. Therefore, segments used for training and testing are produced from separately assigned portions of the original recordings. We note that this protocol is a duration-level split within each original recording, rather than a strict recording-level or vessel-level split.

3.2. Experimental Setup and Hyperparameters

All models are trained under the same optimization protocol to ensure fair comparison. We use the AdamW optimizer with a WarmupCosine learning-rate schedule, including a 10-epoch warmup period to stabilize early training. The batch size, learning rate, and number of epochs are listed in Table 2. The reported recognition accuracy is averaged over the last five epochs to reduce the influence of transient fluctuations.

Table 2. Training hyperparameters.

Epochs	Learning Rate	Batch Sizes
60	1×10^{-4}	32

3.3. Experimental Results

Comparative experiments are conducted to evaluate whether learnable-parameter front-ends provide advantages over fixed-parameter feature representations. By combining three feature representations with two deep learning architectures, we analyze both the recognition performance of LTFF and its compatibility with different backbone models.

The Mel-spectrogram [21] is a widely used acoustic representation because it offers a compact time–frequency description and emphasizes low-frequency information. The Wavelet Scattering Transform (WST) [22] provides a more structured multiscale representation through wavelet bandpass filtering and modulus operations. In contrast to these fixed-parameter front-ends, LTFF learns its filter responses and normalization behavior directly from the target-recognition objective.

To assess LTFF, we combine Mel, WST, and LTFF features with ResNet18 and ViT. Audio Spectrum Transformer (AST) [23] and InceptionNeXt [24] are included as additional reference models. All comparisons are conducted under consistent preprocessing and training settings.

To ensure fair comparison, all baseline methods are implemented under consistent training settings. For Mel-spectrogram and Wavelet Scattering Transform (WST), we set the window length to 512 samples and the hop size to 256 samples. For AST and InceptionNeXt, we use their official base model configurations as released in the corresponding repositories. All backbone networks, including ResNet18, ViT, AST, and InceptionNeXt, are initialized with publicly available pretrained weights. All models are trained using the same optimization strategy, learning-rate schedule, and data preprocessing pipeline.

Comparison for fixed filters: We further evaluate LTFF-ResNet with different numbers of filters on ShipsEar, as summarized in Table 3. The relatively small performance variation suggests that the proposed front-end is not overly sensitive to the tested filter-bank size. Nevertheless, filter number mainly controls representation capacity and computational cost.

Table 3. ResNet-LTFF with varying numbers of filters.

Number of Filters	Accuracy (%)
64	97.3
128	96.9
256	97.1

As shown in Table 4, LTFF-ResNet achieves the highest recognition accuracy on ShipsEar, demonstrating that the learnable frequency-guided front-end can provide discriminative information when coupled with a convolutional backbone. However, LTFF-ViT does not consistently outperform fixed-parameter features. This result indicates a compatibility issue between highly flexible learnable front-ends and Transformer architectures under limited acoustic training data. Fixed front-ends such as Mel and WST inject stronger signal-processing priors, which can partially compensate for the weak locality bias of ViT.

Table 4. Recognition results.

Methods	ShipsEar	DeepShip
LTFF-ResNet	97.3	94.9
LTFF-ViT	93.5	93.9
Mel-ResNet	94.7	90.3
Mel-ViT	95.5	91.0
Wavelet Scatter-ResNet	95.7	91.2
Wavelet Scatter-ViT	96.0	92.3
Mel-AST	95.9	88.1
InceptionNeXt	93.4	86.6

These observations suggest that the benefit of a learnable acoustic front-end is architecture-dependent. LTFF is better aligned with convolutional models because both Gabor filtering and CNN operations emphasize local time–frequency structures [15,16]. Transformer-based models [17,18] may require larger datasets, acoustic pretraining, or locality-aware patch designs to fully exploit LTFF features.

Ablation experiments: Table 5 presents the component-level comparison of LTFF and LEAF under different backbone architectures. Compared with LEAF, LTFF improves the recognition accuracy from 96.2% to 97.3% on ShipsEar and from 92.9% to 94.9% on DeepShip when using ResNet, indicating that the frequency-guided partition provides additional discriminative information beyond the standard learnable front-end. A similar improvement is observed for the ViT backbone, where LTFF achieves 93.5% and 93.9% on ShipsEar and DeepShip, respectively.

Replacing PCEN with conventional logarithmic compression consistently reduces performance for both backbones. For ResNet, the accuracy decreases by 0.6 percentage points on ShipsEar and 0.7 percentage points on DeepShip; for ViT, the corresponding decreases are 0.5 and 0.7 percentage points. This result indicates that adaptive PCEN contributes to the learned representation beyond fixed logarithmic compression. In contrast, removing the squared-modulus operation prevents stable convergence, because the complex Gabor responses can no longer be converted into the real-valued energy representation

required by the subsequent processing pipeline. Overall, the results support the combined role of frequency-guided partitioning, squared-modulus energy extraction, and adaptive PCEN in LTFF.

Table 5. Component-level comparison recognition results.

Method	Frequency-Guided Partition	Squared Modulus	Normalization	ShipsEar	DeepShip
LTFF-ResNet	✓	✓	PCEN	97.3	94.9
LTFF-ViT	✓	✓	PCEN	93.5	93.9
LEAF-ResNet	×	✓	PCEN	96.2	92.9
LEAF-ViT	×	✓	PCEN	92.8	91.3
LTFF-ResNet w/log compression	✓	✓	Log	96.7	94.2
LTFF-ViT w/log compression	✓	✓	Log	93.0	93.2
LTFF-ResNet w/squared modulus	✓	×	PCEN	-	-
LTFF-ResNet w/squared modulus	✓	×	PCEN	-	-

More detailed performance analysis: To further evaluate the effectiveness and reliability of the proposed method, we conduct additional experiments on the ShipsEar and DeepShip datasets using three different random seeds. The mean and standard deviation of precision, recall, and F1-score are reported in Table 6.

Table 6. Comparison recognition results.

Methods	Precision	ShipsEar Recall	F1-Score	Precision	DeepShip Recall	F1-Score
LTFF-ResNet	96.9 ± 0.2	98.2 ± 0.1	97.4 ± 0.2	94.3 ± 0.2	94.6 ± 0.2	94.4 ± 0.3
LTFF-ViT	93.0 ± 0.2	95.6 ± 0.2	94.9 ± 0.3	93.8 ± 0.2	94.0 ± 0.2	94.0 ± 0.3

The results show that LTFF-ResNet achieves a 97.4 ± 0.2 F1-score on ShipsEar and 94.4 ± 0.3 on DeepShip, indicating consistent performance across different runs. LTFF-ViT also demonstrates stable behavior with low variance, further confirming the robustness of the proposed feature representation. In addition, as shown in Table 7, LTFF-ResNet consistently outperforms LEAF-ResNet on both datasets. LTFF achieves a mean F1-score improvement of 1.23 percentage points on ShipsEar and 1.56 percentage points on DeepShip. The paired t-test results indicate that these improvements are statistically significant, with ($p = 0.000365$) on ShipsEar and ($p = 0.000236$) on DeepShip. These results support that the frequency-guided LTFF front-end provides a stable and statistically reliable improvement over the LEAF baseline.

Table 7. Statistical significance results.

Dataset	LTFF-Resnet	LEAF-Resnet	Δ	t	df	p-Value
ShipsEar	97.4 ± 0.2	96.2 ± 0.2	+1.23	37.00	2	0.000365
DeepShip	94.4 ± 0.3	92.9 ± 0.3	+1.56	46.00	2	0.000236

In addition, the confusion matrices provide a fine-grained view of classification behavior. As shown in Figure 2, ShipsEar exhibits nearly perfect diagonal dominance, indicating near-complete class separability. For DeepShip, most classes are well distinguished, while

minor confusion occurs between Cargo and Tanker due to their similar acoustic characteristics. Overall, the results confirm that the proposed method achieves both high accuracy and strong class-wise discrimination.

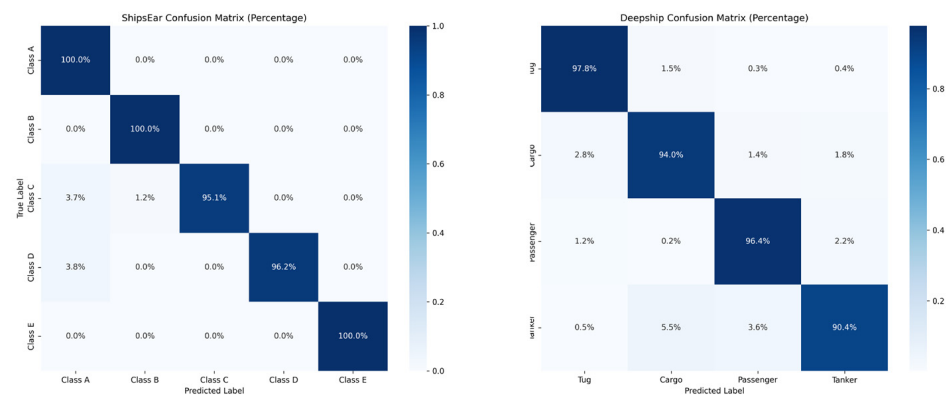


Figure 2. Confusion matrix.

Training loss: Figure 3 compares the training losses of ResNet-LTFF, ResNet-Mel, and ResNet-WST on ShipsEar. The smoother convergence of LTFF indicates that the learnable front-end can provide a more optimization-friendly representation for the convolutional recognizer. However, training loss alone is insufficient to prove generalization; it should be interpreted together with test accuracy and class-wise metrics.

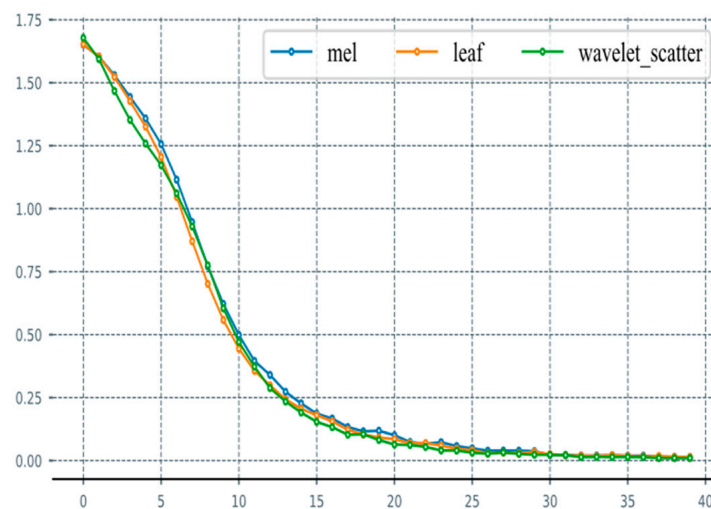


Figure 3. Training loss curves.

Robustness analysis: We evaluate the robustness of the proposed method by adding additive Gaussian white noise to the ShipsEar dataset to construct data with different signal-to-noise ratio (SNR) levels. The performance of LTFF-ResNet, LTFF-ViT, and the Mel-AST baseline is then compared under these noisy conditions. As illustrated in Figure 4, the proposed learnable signal feature extraction method consistently achieves higher accuracy than fixed-feature baselines across all SNR levels. The performance gap becomes more pronounced in the low-SNR regime (−5 dB to 5 dB), where the acoustic signals are heavily corrupted by noise and robust feature representations are critical. This result demonstrates that the proposed LTFF can adaptively extract discriminative features under noisy conditions, leading to more stable and reliable classification performance compared to handcrafted or fixed front-end feature representations. These results indicate improved tolerance to additive Gaussian white noise under the evaluated SNR levels, as depicted

in Figure 4. They do not constitute a comprehensive evaluation of robustness to realistic underwater ambient noise or propagation-related distortions.

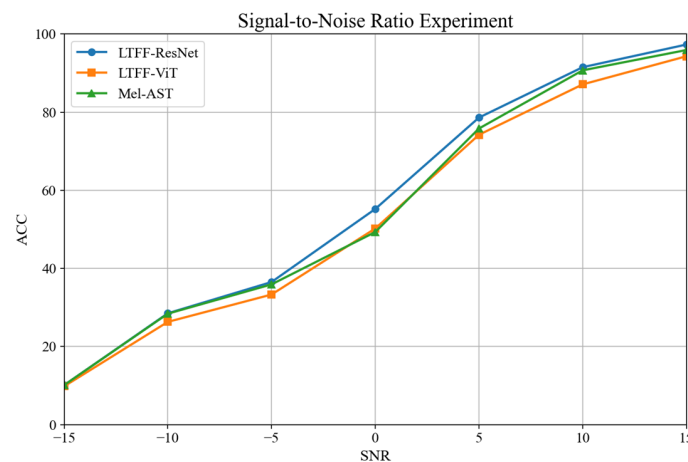


Figure 4. Performance under different SNR levels.

Computational cost and runtime analysis: The results are shown in Table 3. All results are measured on an NVIDIA A6000 (NVIDIA Corporation, Santa Clara, USA) with batch size = 1 under FP32 precision. Input audio length is 3 s at 12 kHz sampling rate. LEAF frontend uses 64 mel-like learnable filter banks with window length 512 and hop size 256. This indicates that in practical deployment scenarios, front-end computation is the primary bottleneck, and FLOPs alone is not sufficient to reflect real inference efficiency.

On an NVIDIA A6000 GPU with 48 GB of memory, we evaluated the training cost of LIFT-based models using a batch size of 16. For ShipsEar, with 3 s input segments, LIFT + ResNet18 required approximately 1.71 GB of peak GPU memory and took about 26.3 s (0.0073 h) per training epoch, whereas LIFT + ViT-Tiny required approximately 1.48 GB and took about 27.6 s (0.0077 h) per epoch. For DeepShip, with 20 s input segments, LIFT + ResNet18 required approximately 10.14 GB of peak GPU memory and took about 419.5 s (0.117 h) per epoch, while LIFT + ViT-Tiny required approximately 9.29 GB and took about 389.1 s (0.108 h) per epoch. Overall, both configurations can be trained on a single NVIDIA A6000 GPU; the peak memory usage remains below 4% of the total GPU memory for ShipsEar and approximately 19–21% for DeepShip. The computational cost and runtime analysis is shown in Table 8.

Table 8. Computational cost and runtime analysis. All results are measured on an NVIDIA A6000 with batch size = 1 under FP32 precision. Input audio length is 3 s at 12 kHz sampling rate. LEAF frontend uses 64 mel-like learnable filter banks with window length 512 and hop size 256.

Pipeline	Params (M)	Flops (G)	LIFT Flops	Backend Flops	End to End Latency (ms, A6000)	LIFT Latency (ms, A6000)	Backend Latency (ms, A6000)
LIFT + ResNet	4.726	4.808	2.369	2.439	9.16	6.39	2.40
LIFT + Vit	5.395	2.551	2.369	2.183	9.64	6.33	2.90

To further examine feature separability, t-SNE visualization [25] is used to project learned representations into a two-dimensional space, as shown in Figure 5 for ShipsEar. In the LTF-ResNet configuration, intra-class samples show relatively compact distributions and inter-class samples show clearer separation, which is consistent with the recognition results. Since t-SNE is qualitative and sensitive to hyperparameters, future work should

complement this visualization with quantitative clustering indicators such as the silhouette score or Davies–Bouldin index.

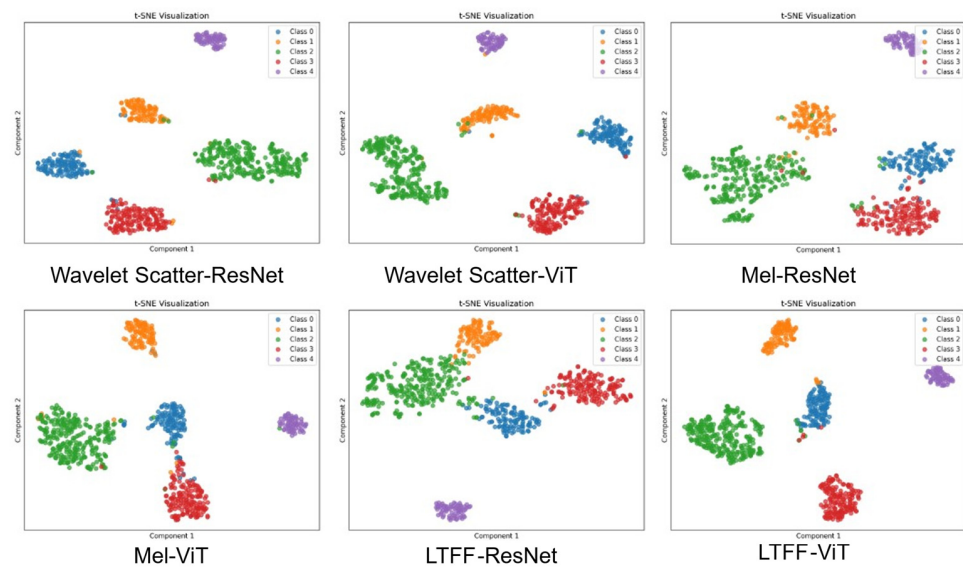


Figure 5. t-SNE visualization on ShipsEar.

The t-SNE visualization results for DeepShip are shown in Figure 6. Because DeepShip contains more vessels, longer recordings, and more variable sea states, the learned feature distributions are less compact than those on ShipsEar. This phenomenon reinforces the need for robustness tests under controlled SNR degradation and for recording-level validation protocols. Overall, the visualizations indicate that the interaction between front-end representation and model architecture substantially affects both separability and stability.

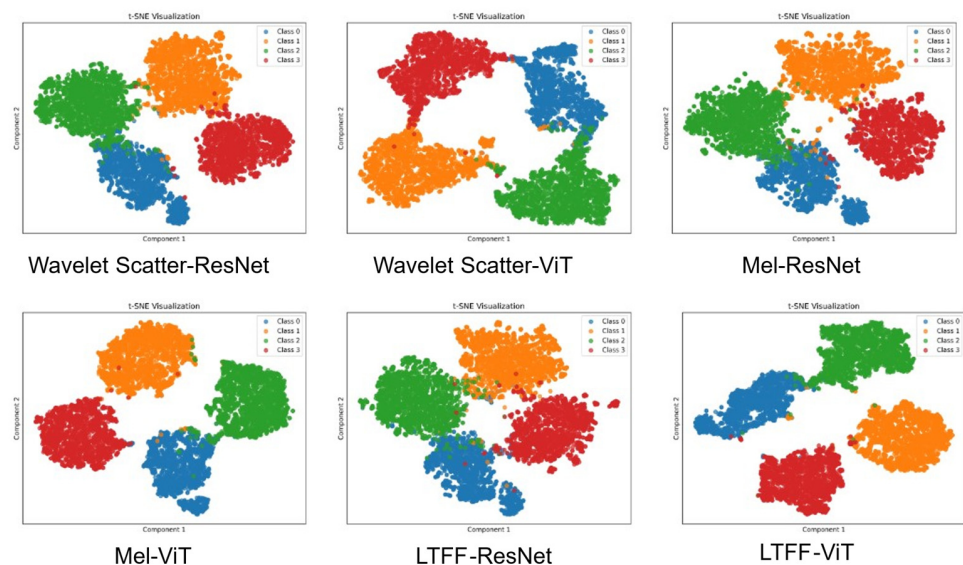


Figure 6. t-SNE visualization on DeepShip.

4. Conclusions

This paper proposes LTFF, a Learnable Two-Stage Frequency-Guided Front-End for underwater acoustic signal recognition. Unlike conventional fixed time–frequency pre-processing methods, LTFF formulates acoustic feature extraction as a differentiable and trainable front-end that can be jointly optimized with the downstream recognition model. Specifically, the proposed method integrates frequency-band partitioning, learnable complex Gabor filtering, squared-modulus envelope extraction, learnable Gaussian pooling,

and per-channel energy normalization into a unified framework. Through this design, LTFF enables the model to adaptively capture frequency-selective target signatures, improve local temporal stability, and suppress slowly varying energy fluctuations in underwater acoustic recordings. The discrete formulation further clarifies the signal-processing flow, trainable parameters, and optimization path of LTFF, thereby enhancing the theoretical consistency and reproducibility of the proposed method.

The experimental results show that LTFF achieves promising recognition performance and exhibits stronger compatibility with convolutional architectures than with vanilla Transformer architectures under the current dataset scale. This architecture-dependent behavior can be explained by the alignment between the localized sub-band energy representations produced by LTFF and the local inductive bias of CNNs, including local connectivity, weight sharing, and hierarchical feature aggregation. In contrast, ViT relies more heavily on global self-attention and weaker locality priors, which may require larger acoustic datasets, stronger pretraining, or locality-aware patch designs to fully exploit the proposed learnable front-end. These findings indicate that the effectiveness of learnable acoustic front-ends depends not only on the signal representation itself, but also on its compatibility with the downstream recognition architecture.

Overall, LTFF provides a task-adaptive alternative to fixed acoustic front-ends for underwater acoustic signal recognition and offers a useful perspective for bridging physically meaningful signal preprocessing with deep representation learning. Future work will focus on more comprehensive validation, including component-level ablation studies, recording-level split verification, low-SNR robustness testing, class-wise performance analysis, and computational-cost evaluation. These studies will further assess the practical value of LTFF for real-world underwater monitoring and embedded deployment scenarios.

5. Limitations

Although LTFF provides a robust learnable front-end under the evaluated conditions, its generalization to unseen vessels, acoustic environments, and recording conditions has not been fully established. Performance may degrade when target characteristics or propagation environments differ substantially from those represented in the training data. Moreover, the current robustness evaluation is limited to controlled additive Gaussian white noise and does not cover realistic ambient noise, reverberation, or propagation-related distortions. Addressing these limitations will require dedicated supervision and training strategies, including vessel-level or environment-level annotations, domain-aware data partitioning, and cross-domain adaptation or generalization methods.

Author Contributions: Conceptualization, H.Z. and H.W.; Software, H.Z.; Methodology, H.Z., S.Z. and H.W.; Validation, H.W., S.Z. and Y.Y.; Project administration, H.W. and Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Natural Science Foundation of China grant No. 62031021 and 62271404.

Data Availability Statement: Data and program will be made available on request.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Duarte, C.M.; Chapuis, L.; Collin, S.P.; Costa, D.P.; Devassy, R.P.; Eguiluz, V.M.; Erbe, C.; Gordon, T.A.C.; Halpern, B.S.; Harding, H.R.; et al. The soundscape of the Anthropocene ocean. *Science* **2021**, *371*, eaba4658. [CrossRef]
2. Vaccaro, R.J. The past, present, and the future of underwater acoustic signal processing. *IEEE Signal Process. Mag.* **1998**, *15*, 21–51. [CrossRef]

3. Doan, V.-S.; Huynh-The, T.; Kim, D.-S. Underwater acoustic target classification based on dense convolutional neural network. *IEEE Geosci. Remote Sens. Lett.* **2020**, *19*, 1500905. [\[CrossRef\]](#)
4. Wang, B.; Zhang, W.; Zhu, Y.; Wu, C.; Zhang, S. An underwater acoustic target recognition method based on AMNet. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 5501105. [\[CrossRef\]](#)
5. Miao, Y.; Zakharov, Y.V.; Sun, H.; Li, J.; Wang, J. Underwater acoustic signal classification based on sparse time–frequency representation and deep learning. *IEEE J. Ocean. Eng.* **2021**, *46*, 952–962. [\[CrossRef\]](#)
6. Liu, F.; Shen, T.; Luo, Z.; Zhao, D.; Guo, S. Underwater target recognition using convolutional recurrent neural networks with 3-D Mel-spectrogram and data augmentation. *Appl. Acoust.* **2021**, *178*, 107989. [\[CrossRef\]](#)
7. Feng, S.; Zhu, X. A transformer-based deep learning network for underwater acoustic target recognition. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1505805. [\[CrossRef\]](#)
8. Zhou, A.; Li, X.; Zhang, W.; Zhao, C.; Ren, K.; Ma, Y.; Song, J. An attention-based multi-scale convolution network for intelligent underwater acoustic signal recognition. *Ocean Eng.* **2023**, *287*, 115784. [\[CrossRef\]](#)
9. You, K.; Xu, K.; Feng, M.; Zhu, B. Underwater acoustic classification using masked modeling-based swin transformer. *J. Acoust. Soc. Am.* **2022**, *152*, A296. [\[CrossRef\]](#)
10. Zeghidour, N.; Teboul, O.; Quitry, F.D.C.; Tagliasacchi, M. LEAF: A learnable front-end for audio classification. *arXiv* **2021**, arXiv:2101.08596.
11. Fu, Q.; Teng, Z.; White, J.; Powell, M.E.; Schmidt, D.C. Fastaudio: A learnable audio front-end for spoof speech detection. In *ICASSP 2022—2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2022; pp. 3693–3697.
12. Verma, P.; Chafe, C. A content adaptive learnable “time-frequency” representation for audio signal processing. In *ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–5.
13. Ren, J.; Xie, Y.; Zhang, X.; Xu, J. UALF: A learnable front-end for intelligent underwater acoustic classification system. *Ocean Eng.* **2022**, *264*, 112394. [\[CrossRef\]](#)
14. Xie, Y.; Ren, J.; Xu, J. Adaptive ship-radiated noise recognition with learnable fine-grained wavelet transform. *Ocean Eng.* **2022**, *265*, 112626. [\[CrossRef\]](#)
15. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: Analysis, applications, and prospects. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 6999–7019. [\[CrossRef\]](#) [\[PubMed\]](#)
16. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A Survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 87–110. [\[CrossRef\]](#) [\[PubMed\]](#)
17. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
18. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.J. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
19. Santos-Domínguez, D.; Torres-Guijarro, S.; Cardenal-López, A.; Pena-Gimenez, A. ShipsEar: An underwater vessel noise database. *Appl. Acoust.* **2016**, *113*, 64–69. [\[CrossRef\]](#)
20. Irfan, M.; Jiangbin, Z.; Ali, S.; Iqbal, M.; Masood, Z.; Hamid, U. DeepShip: An underwater acoustic benchmark dataset and a separable convolution based autoencoder for classification. *Expert Syst. Appl.* **2021**, *183*, 115270. [\[CrossRef\]](#)
21. Koppurapu, S.K.; Laxminarayana, M. Choice of Mel filter bank in computing MFCC of a resampled speech. In *2010 10th International Conference on Information Sciences, Signal Processing and their Applications (ISSPA)*; IEEE: Piscataway, NJ, USA, 2010; pp. 121–124.
22. Anden, J.; LOSTANLEN, V.; Mallat, S. Joint time–frequency scattering. *IEEE Trans. Signal Process.* **2019**, *67*, 3704–3718. [\[CrossRef\]](#)
23. Gong, Y.; Chung, Y.-A.; Glass, J. Ast: Audio spectrogram transformer. *arXiv* **2021**, arXiv:2104.01778.
24. Yu, W.; Zhou, P.; Yan, S.; Wang, X. Inceptionnext: When inception meets convnext. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Piscataway, NJ, USA, 2024; pp. 5672–5683.
25. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.