Journal of Computer Science and Digital Technology, 2026, 2(2), 30-50
https://doi.org/10.71204/6bsh1667

Ultrasonic Signal Denoising Based on One-Dimensional Convolution and SWT-MRA Multi-Scale Representation

Weijiang Qiao [1], Yuhan Wang [1], Guanghan Ding [1], Yanlong Wei [1,*]

[1] College of Computer Science and Technology, Taiyuan Normal University, Taiyuan 030619, China

*** Correspondence:**

Yanlong Wei

weiyanlong@tynu.edu.cn

Received: 12 October 2026 / Accepted: 10 November 2026 / Published online: 11 November 2026

Abstract

Ultrasonic measurement signals are susceptible to random noise, impulsive artifacts, and high-frequency interference, while conventional wavelet thresholding may attenuate useful signal components. To address the trade-off between noise suppression and waveform preservation, this study proposes an ultrasonic signal denoising method that integrates stationary wavelet transform-based multiresolution analysis (SWT-MRA) with a one-dimensional convolutional neural network (CNN). An eight-level SWT-MRA decomposition using the sym8 wavelet produces one approximation component and eight equal-length detail components. A dilated residual CNN then learns a component-domain denoising mapping under joint waveform-, frequency-, gradient-, and component-level constraints. The method was evaluated on three independent experimental batches, and the learning-based models were repeatedly trained using five random seeds. On the independent test set, the proposed method achieved a mean absolute error (MAE) of 0.003685, a root mean square error (RMSE) of 0.005024, and an output signal-to-noise ratio (SNR) of 22.198 dB. Compared with the time-domain CNN, it reduced MAE and RMSE by 3.10% and 3.39%, respectively, while increasing output SNR by 0.297 dB. Compared with 1D U-Net, MAE and RMSE were reduced by 9.22% and 10.19%, respectively. Ablation experiments showed that the main performance gain was attributable to the explicit SWT-MRA multiscale representation, whereas component-level auxiliary supervision provided limited additional benefit. These results demonstrate that explicit multiscale representation can improve ultrasonic waveform recovery with only a modest increase in network size.

Keywords: Ultrasonic Signal Denoising; Stationary Wavelet Transform; Multiresolution Analysis; One-Dimensional Convolutional Neural Network; Multiscale Representation

1. Introduction

Ultrasonic nondestructive testing and thermometry rely on waveform features such as arrival time, amplitude, and local shape (Tsai et al., 2006; Schmidt et al., 2014). At high temperatures or in the presence of strong electromagnetic interference and complex data-acquisition chains, ultrasonic measurements can be affected by electronic noise, environmental disturbances, impulsive interference, and propagation attenuation. These effects may introduce high-frequency oscillations, spikes, and baseline fluctuations near echoes of interest. Because arrival-time estimation, echo matching, and temperature inversion depend on these echoes, denoising must suppress interference while preserving their amplitudes and temporal positions (Schmidt et al., 2014; Abbate et al., 1997).

Common approaches to ultrasonic signal denoising include time- and frequency-domain filtering, averaging repeated measurements, and wavelet thresholding. Unlike fixed-bandwidth filters, wavelet transforms represent local features of nonstationary signals at multiple scales and are widely used for transient-signal analysis (Abbate et al., 1997; Mallat, 1989; Nason and Silverman, 1995). The discrete wavelet transform (DWT) produces approximation and detail coefficients through successive filtering and downsampling (Mallat, 1989), and noise can be reduced by thresholding these coefficients (Donoho, 1995). However, downsampling makes the DWT sensitive to signal shifts (Nason and Silverman, 1995). In addition, fixed or statistically selected thresholds may not reliably separate signal-related coefficients from noise across scales and acquisition conditions. Excessive thresholding can therefore attenuate useful coefficients and distort local waveforms (Donoho, 1995; Lázaro et al., 2002; Pardo et al., 2006).

Deep learning has increasingly been applied to ultrasonic nondestructive evaluation and denoising (Cantero-Chinchilla et al., 2022). Generative adversarial networks, denoising autoencoders, and physics-informed deep filtering have been investigated for ultrasonic and guided-wave signal processing (Cui et al., 2022; Singh et al., 2023; Li et al., 2023; Jadhav et al., 2023; Wu et al., 2025). Recent work has also explored improved U-Net architectures for ultrasonic echo denoising and wavelet-initialized attention U-Net models for guided-wave restoration (Zhang et al., 2026; Qian et al., 2026). The original U-Net architecture was developed for biomedical image segmentation (Ronneberger et al., 2015). Learned models reduce the need for hand-crafted thresholding rules, but networks operating directly on single-channel waveforms must learn to separate low-frequency signal structure from high-frequency disturbances implicitly. Greater depth and encoder–decoder designs can enlarge the receptive field at the cost of additional architectural or computational complexity, without necessarily improving performance on a given ultrasonic denoising task.

We therefore combine SWT-MRA with a one-dimensional residual CNN. By avoiding downsampling, SWT maintains the original signal length at every scale. MRA reconstructs these coefficients as additive time-domain components, yielding a multichannel input with explicit scale-specific information. Rather than applying fixed thresholds to wavelet detail coefficients, the convolutional network jointly estimates denoised components, with an optional component-level auxiliary loss during training. The principal contributions of this study are as follows:

(1) We construct an equal-length multiscale representation using SWT-MRA and combine it with a capacity-matched one-dimensional dilated residual CNN, preserving temporal resolution while explicitly exploiting information at different scales.

(2) Using a common data split, training protocol, and model-selection criterion, we compare the time-domain CNN, SWT-MRA-CNN without component-level supervision, and the full SWT-MRA-CNN. Five random seeds and 45 sensor–temperature groups are used to assess the contributions of multiscale input and auxiliary component supervision.

(3) We compare the proposed method with conventional SWT soft thresholding, a time-domain CNN, and 1D U-Net on an independent experimental batch. Performance is assessed through quantitative errors, waveform visualizations, model complexity, GPU-only forward-pass latency, and end-to-end inference latency.

2. Data and Problem Definition

2.1. Ultrasonic Signal Acquisition and Dataset Composition

A denoising dataset was assembled from repeated ultrasonic measurements obtained with three sensors, 15 temperature conditions, and three independent experimental batches. The temperatures comprised room temperature and 14 elevated-temperature points between 300 and 1600 °C. For each sensor–temperature–batch combination, five repeated measurements were acquired. Each batch therefore contained $3 \times 15 \times 5 = 225$ waveforms, giving 675 waveforms in total. Every waveform comprised 7,680 samples recorded at 40 MHz, corresponding to a sampling interval of 0.025 μs.

Table 1. Dataset composition

Item	Setting
Number of sensors	3
Temperature conditions	15
Number of repetitions for each condition	5
Independent experimental batch	3
Sample size of a single experimental batch	225
Total sample size	675
Single signal length	7680 sampling points
sampling frequency	40 MHz

Because measurements from the same sensor and temperature were repeated within each batch, individual repetitions were not randomly distributed between the training and test sets. Instead, the data were separated by experimental batch. This prevents batch-specific measurement effects from being shared between training and testing and provides a stricter evaluation of cross-batch generalization.

2.2. Construction of Leave-One-Out Averaged Reference Signals

Because physically noise-free reference waveforms are unavailable, a leave-one-out mean is constructed from five repeated measurements acquired under identical sensor, temperature, and experimental batch conditions. For repetition r, the corresponding waveform is treated as the noisy input and the other four aligned waveforms are averaged. Let k denote the waveform after alignment for repetition $x_k[n]$, and let the total number of repetitions be $R = 5$. The reference for repetition r is defined as:

$$y_r[n] = \frac{1}{R-1}\sum_{k \neq r} x_k[n], \quad R = 5 \tag{1}$$

Averaging the other four measurements reduces random noise without incorporating the target repetition into its own reference. Common interference and batch-specific biases may nevertheless remain in the average. Therefore, $y_k[n]$ is a leave-one-out averaged, pseudo-clean reference rather than a physically noise-free ground truth. All reported errors, correlations, and SNR measures are calculated relative to this reference.

2.3. Signal Preprocessing

A common preprocessing procedure was applied to all waveforms to mitigate baseline offsets, small temporal misalignments, and amplitude outliers. It comprised baseline correction, within-group temporal alignment, and robust normalization. Alignment was performed only within a given experimental batch and sensor–temperature group; waveforms from different data splits were never aligned with one another.

(1) Baseline correction. For each waveform, the mean of samples with indices 0–2499 was subtracted from the complete signal to reduce the DC offset and slow baseline variations. (2) Temporal alignment. Relative shifts were estimated over samples with indices 2500–7399, which contain the principal echo region, using the five repeated measurements within each sensor–temperature–batch group. Shifts were limited to 20 samples, and two alignment iterations were performed to align the main echoes. (3) Robust normalization. The 99.5th percentile of absolute amplitude was used to define a robust scaling factor that is less sensitive to isolated spikes. The global scaling statistic was computed exclusively from the training set and applied unchanged to the validation and test sets, preventing test-set statistics from informing training. Each preprocessed waveform retained its original length of 7,680 samples. The SWT-MRA model subsequently formed nine multiscale input channels, with per-component scale normalization performed during model input construction, as described in Section 3.

2.4. Denoising Problem Formulation

Let the preprocessed noisy waveform be $x \in R^N$ and the leave-one-out reference be $y \in R^N$, with N = 7,680 samples. We seek a parameterized mapping F_θ such that the predicted waveform $\hat{y}$ approximates the reference y:

$$\hat{\mathbf{y}} = F_\theta(\mathbf{x}) \tag{2}$$

For the time-domain CNN and 1D U-Net, F_θ takes a single-channel waveform as input. For SWT-MRA-CNN, the waveform x is first represented by nine equal-length multiscale components using the fixed wavelet transform T_MRA. The one-dimensional CNN then predicts denoised components, which are summed after their amplitude scales have been restored. The overall mapping can be written as:

$$\hat{\mathbf{y}} = R\left(G_\theta\left(T_{\mathrm{MRA}}(\mathbf{x})\right)\right) \tag{3}$$

Here, G_θ represents the trainable one-dimensional CNN, and $R(\ )$ represents amplitude restoration and summation across the multiscale components. We investigate whether explicit SWT-MRA representation improves waveform recovery while maintaining a comparable CNN size.

3. Proposed SWT-MRA-CNN Method

3.1. Overall Architecture

The proposed framework comprises SWT-MRA representation, one-dimensional convolutional denoising in the component domain, and waveform reconstruction (Figure 1). First, the preprocessed single-channel signal undergoes an eight-level SWT-MRA decomposition using the sym8 wavelet, producing nine time-domain components of the same length as the input. Each component is normalized by its root mean square (RMS), and the components are stacked to form a 9 × 7,680 input tensor. A one-dimensional CNN with dilated residual blocks jointly models information across scales and predicts nine denoised components. Their amplitudes are restored and the components are summed to reconstruct the output waveform. Training constrains both the reconstructed waveform and intermediate multiscale predictions; inference requires only SWT-MRA construction, network evaluation, and component reconstruction.

3.2. SWT Decomposition and MRA Representation

The DWT performs multiresolution signal decomposition using low-pass and high-pass filter banks (Mallat, 1989). Because the conventional DWT downsamples at each level, coefficient lengths decrease with decomposition depth and the resulting representation is shift-sensitive. SWT omits downsampling, retaining coefficients aligned with the original time axis and improving shift invariance (Nason and Silverman, 1995). Such undecimated wavelet representations have also been applied to noise suppression in ultrasonic nondestructive testing (Pardo et al., 2006).

The sym8 wavelet and eight-level SWT are used throughout this study. Multiresolution reconstruction converts the coefficients into one approximation component and eight detail components, denoted m_0, m_1, …, m_8. Each component is an element of R^N and the components satisfy the approximate perfect-reconstruction relation:

$$x[n] \approx \sum_{j=0}^{8} m_j[n] \tag{4}$$

A numerical consistency check used the same SWT-MRA procedure as training to regenerate the multiscale components. The regenerated and stored components agreed exactly, and the maximum absolute error after component summation was of the order of 10^{-7}. This confirms the numerical consistency of the implementation. Unlike DWT coefficients, which have scale-dependent lengths, the nine reconstructed SWT-MRA components each contain 7,680 samples and can be stacked along the channel dimension (Nason and Silverman, 1995):

$$\mathbf{M} = \begin{bmatrix} \mathbf{m}_0 \\ \mathbf{m}_1 \\ \vdots \\ \mathbf{m}_8 \end{bmatrix} \in R^{9\times N} \tag{5}$$

Because the component energies differ substantially across scales, each component of every sample is normalized by its RMS. For scale , let denote the RMS of the corresponding component. It is computed as:

$$s_j = \sqrt{\frac{1}{N}\sum_{n=1}^{N} m_j^2[n]} \tag{6}$$

$$\tilde{m}_j[n] = \frac{m_j[n]}{s_j + \varepsilon} \tag{7}$$

A small constant $\varepsilon = 10^{-6}$ prevents numerical instability for low-energy components. The nine normalized components are stacked as network inputs. Their original amplitude scales are restored before waveform reconstruction.

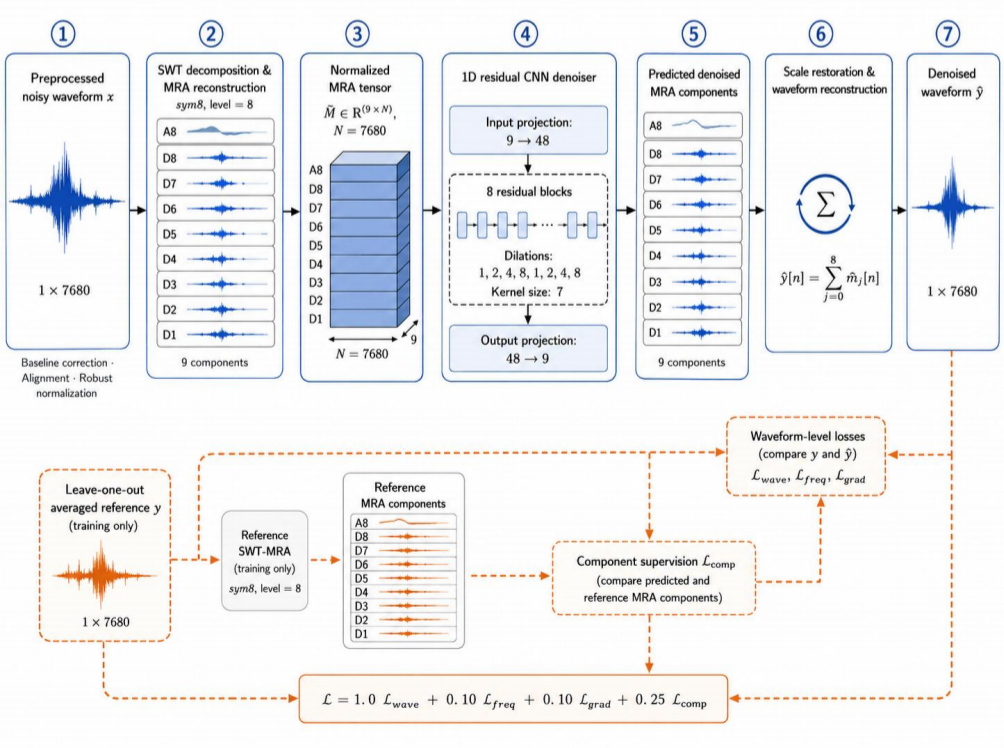


Figure 1. Overall architecture of SWT-MRA-CNN.

3.3. One-Dimensional Convolutional Denoising Network

The component-domain denoiser is a one-dimensional residual CNN that preserves the temporal dimension. It uses 48 base feature channels, eight residual blocks, and a convolutional kernel size of 7. The dilation factors of the eight blocks are 1, 2, 4, 8, 1, 2, 4, and 8. These settings provide both local context around neighboring samples and a wider receptive field for echo structures without pooling or upsampling. Consequently, the waveform length remains unchanged throughout the network.

SWT-MRA-CNN accepts nine input channels and predicts one denoised component for each MRA scale. Multichannel convolution enables each component prediction to exploit information from other scales. The time-domain CNN uses the same base channel count, residual blocks, kernel size, and dilation factors, but operates on a single-channel waveform. The two networks have 261,073 and 266,457 trainable parameters, respectively, a difference of approximately 2.06%; thus, their parameter counts are comparable.

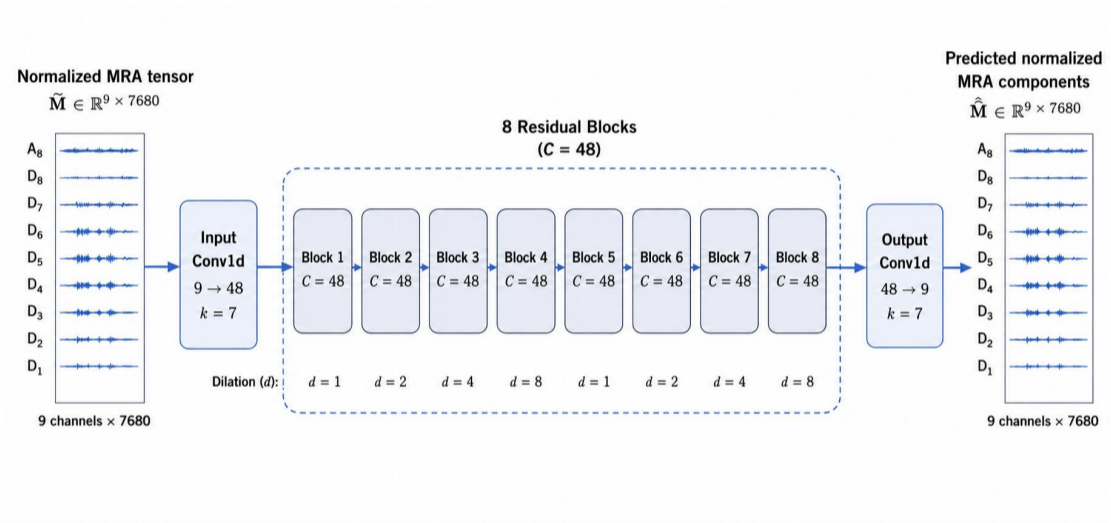


Figure 2. Structure of the one-dimensional dilated residual denoising CNN.

3.4. Multiscale Component Fusion and Waveform Reconstruction

Let the nine denoised components predicted by the network be $\hat{m}_j$. After restoring the original component amplitude scales, the final time-domain waveform is reconstructed by summation:

$$\hat{y}[n] = \sum_{j=0}^{8} \hat{m}_j[n] \tag{8}$$

Reconstruction requires no additional trainable decoder and preserves the original waveform length. The full SWT-MRA-CNN and its variant without component-level supervision use identical SWT-MRA preprocessing, CNN architecture, and reconstruction at inference; consequently, their parameter counts, MACs, weight-file sizes, and memory requirements are identical. They differ only during training: the full model includes component-level auxiliary supervision, whereas the corresponding loss weight is set to zero in the ablation model.

3.5. Loss Functions

The full model uses four weighted losses to constrain waveform accuracy, spectral magnitude, local sample-to-sample variations, and multiscale components. Let the predicted waveform be $\hat{y}$ and its reference be y . Denote the predicted MRA component by $\hat{m}_j$ and the corresponding reference component by m_j^{ref} . The waveform L1 loss penalizes pointwise temporal errors:

$$L_{\text{wave}} = \frac{1}{N}\sum_{n=1}^{N}\left|\hat{}[n] - [\ ]\right| \tag{9}$$

The frequency-domain L1 loss penalizes differences between the magnitudes of the predicted and reference spectra:

$$L_{\text{freq}} = \frac{1}{K}\sum_{k=1}^{K}\left|\left|F(\hat{\mathbf{y}})_k\right| - \left|F(\mathbf{y})_k\right|\right| \tag{10}$$

where F(·) denotes the discrete Fourier transform and K is the number of frequency bins included in the loss.

The gradient-domain L1 loss penalizes differences between adjacent-sample variations, helping to preserve local oscillatory structure:

$$L_{\text{grad}} = \frac{1}{N-1}\sum_{n=1}^{N-1}\left|\left(\hat{y}[n+1]-\hat{y}[n]\right) - \left(y[n+1]-y[n]\right)\right| \tag{11}$$

The component-level auxiliary loss penalizes deviations between the predicted and reference components across the nine MRA scales:

$$L_{\text{comp}} = \frac{1}{9N}\sum_{j=0}^{8}\sum_{n=1}^{N}\left|\hat{m}_j[n] - m_j^{\text{ref}}[n]\right| \tag{12}$$

The overall training objective is:

$$L = \lambda_{\text{wave}}L_{\text{wave}} + \lambda_{\text{freq}}L_{\text{freq}} + \lambda_{\text{grad}}L_{\text{grad}} + \lambda_{\text{comp}}L_{\text{comp}} \tag{13}$$

The weights of the waveform, frequency, gradient, and component losses were set to 1.0, 0.10, 0.10, and 0.25, respectively.

For SWT-MRA-CNN without component-level supervision, only the component-loss weight L_{comp} is set to zero; all other training settings remain unchanged. Component supervision provides an additional training constraint, but whether it improves waveform recovery must be established through ablation rather than assumed from its formulation.

4. Experimental Setup

We evaluated SWT-MRA-CNN against noisy input, SWT soft thresholding, a one-dimensional time-domain CNN, and 1D U-Net under consistent data partitions, preprocessing, training settings, and evaluation criteria. Five-seed experiments, ablation studies, comparisons across sensors and

temperatures, waveform visualizations, and complexity measurements were used to assess denoising accuracy, stability, and computational cost.

4.1. Dataset Partitioning and Evaluation Protocol

The three experimental batches were assigned to training (batch 1), validation (batch 2), and final independent testing (batch 3). The validation batch was used for learning-rate scheduling, hyperparameter and threshold-rule selection, and checkpoint selection. Each split contained 225 waveforms from three sensors and 15 temperature conditions, with five repetitions per condition. The test set was excluded from all training, threshold selection, and hyperparameter tuning.

All methods used the same preprocessed signals and leave-one-out averaged reference targets defined in Section 2. Because the same three sensors appear in all three splits, this protocol evaluates cross-batch generalization rather than performance on previously unseen sensors.

4.2. Experimental Environment and Training Parameters

Experiments were conducted on Windows using Python 3.11 and PyTorch, with an NVIDIA GeForce RTX 3060 Laptop GPU for neural-network training and inference. The reported complexity and inference benchmarks used PyTorch 2.13.0+cu130 and CUDA 13.0. Automatic mixed precision was enabled during training.

All trainable models used AdamW (Loshchilov and Hutter, 2019) with an initial learning rate of 3×10^{-4}, a weight decay of 1×10^{-5}, a batch size of 8, and a maximum of 100 epochs. A ReduceLROnPlateau scheduler monitored validation waveform L1 error; the learning rate was multiplied by 0.5 after five epochs without improvement. Early stopping used a patience of 15 epochs, and gradients were clipped at 5.0. The checkpoint with the lowest validation waveform L1 error was retained.

Table 2. Experimental and training settings

Parameter	Setting
Training / Validation / Testing	Experimental batches 1/2/3
Optimizer /Weight Decay	AdamW/1×10^{-5}
Initial learning rate	3×10^{-4}
Batch size	8
Maximum number of rounds / Early stop patience value	100/15
Optimal Model Selection Indicators	Validation set waveform_l1
Random Seed	42, 2024, 2025, 2026, 3407
SWT Mother Wavelet / Number of Layers	sym8/8
MRA input channel count	9

Each deep learning model was trained with five random seeds: 42, 2024, 2025, 2026, and 3407. Results are reported as the mean and sample standard deviation. Training-data shuffling was controlled by separately initialized random-number generators, and the output-layer initialization strategy was kept consistent across the relevant model comparisons to minimize confounding differences. Table 2 summarizes the experimental settings.

4.3. Comparison Methods

Six inputs or model configurations were evaluated:

Noisy input. The preprocessed noisy waveform was evaluated without denoising to establish the baseline noise level.

SWT soft thresholding. An eight-level SWT using the sym8 wavelet was applied, followed by soft-threshold shrinkage of the detail coefficients (Donoho, 1995; Lázaro et al., 2002; Pardo et al., 2006). The Universal and Bayes threshold rules were compared using validation-set mean MAE. Universal achieved an MAE of 0.005071, compared with 0.005649 for Bayes, and was therefore fixed for the single evaluation on the test set.

Time-domain CNN. A one-dimensional residual CNN with 48 base channels, eight residual blocks, a kernel size of 7, and dilation factors of 1, 2, 4, 8, 1, 2, 4, and 8. It operates directly on the single-channel waveform.

1D U-Net. A one-dimensional encoder–decoder baseline derived from the U-Net architecture (Ronneberger et al., 2015), with 8 base channels, a depth of 4, and a kernel size of 5. It provides a comparison with a hierarchical multiscale neural architecture.

SWT-MRA-CNN without component-level supervision. This model uses the same nine-channel SWT-MRA input and network architecture as the full model, but its component-level loss weight is set to zero to isolate the effect of multiscale representation.

Full SWT-MRA-CNN. The proposed nine-channel model combines waveform-, frequency-, and gradient-domain losses with component-level auxiliary supervision.

The full and no-component-supervision variants have identical inference architectures, allowing the contribution of the auxiliary loss to be examined separately. The time-domain CNN and both MRA variants share the same main network depth, base channel count, and dilation settings, reducing network-capacity differences in the comparison.

4.4. Evaluation Metrics

Let y, x, and ŷ denote the reference, noisy input, and denoised output, respectively, each with N samples. We evaluate MAE, RMSE, correlation coefficient, output SNR, SNR improvement, peak amplitude error (PAE), and absolute peak shift (APS).

Mean absolute error is defined as :

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{y}_i - y_i\right| \tag{14}$$

The root mean square error is defined as:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2} \tag{15}$$

The correlation coefficient is defined as:

$$\rho = \frac{\sum_{i=1}^{N}\left(\hat{y}_i - \bar{\hat{y}}\right)\left(y_i - \bar{y}\right)}{\sqrt{\sum_{i=1}^{N}\left(\hat{y}_i - \bar{\hat{y}}\right)^2}\sqrt{\sum_{i=1}^{N}\left(y_i - \bar{y}\right)^2}} \tag{16}$$

The output signal-to-noise ratio is defined as:

$$\mathrm{SNR}_{\mathrm{out}} = 10\log_{10}\frac{\sum_{i=1}^{N} y_i^2}{\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2} \tag{17}$$

The input signal-to-noise ratio is calculated using the error between the noisy input and the reference signal.

$$\mathrm{SNR}_{\mathrm{in}} = 10\log_{10}\frac{\sum_{i=1}^{N} y_i^2}{\sum_{i=1}^{N}\left(x_i - y_i\right)^2} \tag{18}$$

The signal-to-noise ratio improvement is:

$$\Delta\mathrm{SNR} = \mathrm{SNR}_{\mathrm{out}} - \mathrm{SNR}_{\mathrm{in}} \tag{19}$$

To assess the preservation of echo amplitude and location, the maximum absolute-amplitude sample was identified separately in each reference and predicted waveform within indices 2500–7399. PAE is the absolute difference between these peak amplitudes; APS is the absolute difference between their sample indices. Lower MAE, RMSE, PAE, and APS indicate better performance, whereas higher correlation, output SNR, and SNR improvement indicate better performance.

4.5. Fairness and Reproducibility

All trainable methods used the same training, validation, and test sets, preprocessing, batch size, initial learning rate, learning-rate scheduler, early-stopping rules, and checkpoint-selection criterion to minimize differences unrelated to the model architecture.

For each seed, a separately initialized random-number generator controlled training-data shuffling to provide the same sample order across models. Checkpoints were selected by the minimum validation waveform L1 error, and no test samples were used for training decisions.

The SWT threshold rule was selected using only validation data. Once Universal thresholding was selected, it was applied to the test set without adjustment. Noisy input and SWT soft thresholding are deterministic baselines, for which one test result is reported. Deep learning methods are summarized by the mean and standard deviation across five seeds.

4.6. Complexity and Inference Timing Protocol

Model complexity was evaluated using the checkpoint for seed 42, an input length of 7,680, and FP32 precision. Parameter counts and model-file sizes were obtained from the network weights. The reported multiply–accumulate operations (MACs) cover only Conv1d and Linear layers; SWT-MRA preprocessing, normalization, activation functions, interpolation, concatenation, and elementwise operations were excluded.

GPU-only forward-pass latency was measured over five runs, each comprising 50 warm-up iterations followed by 200 timed iterations. Latency and throughput were measured at batch sizes of 1 and 8. Input tensors were resident on the GPU during these measurements; file loading and CPU–GPU transfers were excluded. End-to-end inference timing covered all 225 test waveforms and was repeated five times. Measurements began with baseline-corrected, aligned, normalized signals already in memory. Timing included SWT-MRA construction where applicable, CPU–GPU transfer, network inference, waveform reconstruction, and GPU–CPU transfer, but excluded raw-file I/O, plotting, and metric computation.

5. Experimental Results and Discussion

5.1. Overall Quantitative Comparison

Table 3 summarizes test-set performance. Single results are reported for the deterministic baselines, whereas neural-network results are shown as means and standard deviations over five seeds.

Table 3. Test-set performance of different denoising methods

Method	MAE↓	RMSE↓	Correlation coefficient ↑	Output SNR/dB↑	SNR improvement (dB) ↑	Peak amplitude error ↓	Absolute peak shift ↓
Noisy Input	0.006022	0.008548	0.988650	17.561	0.000	0.005224	1.156
SWT Soft Threshold	0.005396	0.007628	0.990931	18.558	0.996	0.006345	0.978
Time-domain CNN	0.003803 ±0.000031	0.005200 ±0.000059	0.995794 ±0.000109	21.901 ±0.094	4.339 ±0.094	0.004930 ±0.000301	0.944 ±0.043
1D U-Net	0.004059 ±0.000044	0.005594 ±0.000103	0.995149 ±0.000154	21.280 ±0.163	3.719 ±0.163	0.004587 ±0.000389	1.019 ±0.050
Full SWT-MRA-CNN	0.003685 ±0.000010	0.005024 ±0.000024	0.996026 ±0.000026	22.198 ±0.046	4.637 ±0.046	0.004172 ±0.000145	0.954 ±0.032

As shown in Table 3, noisy input had an MAE of 0.006022, an RMSE of 0.008548, and an output SNR of 17.561 dB. SWT soft thresholding reduced MAE and RMSE to 0.005396 and 0.007628, respectively, and increased output SNR by 0.996 dB. However, its peak amplitude error increased from 0.005224 to 0.006345 (approximately 21.45%), suggesting that threshold shrinkage can attenuate useful echo amplitudes.

The learning-based models achieved lower overall errors and higher output SNR than SWT soft thresholding. The time-domain CNN obtained an MAE of 0.003803 and an RMSE of 0.005200. The corresponding values for 1D U-Net were 0.004059 and 0.005594. Thus, the encoder–decoder baseline did not outperform the simpler time-domain CNN under the present experimental conditions.

The full SWT-MRA-CNN achieved an MAE of 0.003685, an RMSE of 0.005024, and an output SNR of 22.198 dB. Relative to the time-domain CNN, its MAE and RMSE were lower by 3.10% and 3.39%, output SNR was higher by 0.297 dB, and peak amplitude error was lower by approximately 15.38%. Relative to 1D U-Net, MAE and RMSE were lower by 9.22% and 10.19%, respectively, and output SNR was higher by 0.917 dB. These gains were obtained with a comparable CNN parameter count.

The mean absolute peak shift was 0.954 samples for the full model and 0.944 samples for the time-domain CNN. Although the former was slightly higher, the two values were close. The proposed method therefore provided comparable peak localization while reducing overall waveform error and peak amplitude error.

5.2. Stability Across Random Seeds

To examine sensitivity to initialization, the time-domain CNN, 1D U-Net, and full SWT-MRA-CNN were trained independently with five random seeds. The standard deviations of the full model's MAE, RMSE, and output SNR were 1.0×10^{-5}, 2.4×10^{-5}, and 0.046 dB, respectively, all lower than those of the two reference networks.

The five RMSE values of the full model were 0.005034, 0.005007, 0.004990, 0.005039, and 0.005050. Each was lower than the RMSE of the time-domain CNN trained with the corresponding seed. Thus, the observed improvement was not driven by a single initialization.

Across the 45 sensor–temperature groups, the full model achieved lower MAE and RMSE and higher output SNR than the time-domain CNN in all groups. This consistency across groups complements the five-seed results. However, with only five random seeds, the analysis is descriptive and does not establish statistical superiority through small-sample significance testing.

5.3. Ablation Study

The roles of SWT-MRA input and component-level auxiliary supervision were examined using three configurations: the time-domain CNN, SWT-MRA-CNN without component supervision, and the full SWT-MRA-CNN (Table 4).

Table 4. Five-seed ablation results

Metric	Time-domain CNN	MRA-CNN (no component loss)	Full SWT-MRA-CNN
MAE↓	0.003803±0.000031	0.003684±0.000011	0.003685±0.000010
RMSE↓	0.005200±0.000059	0.005019±0.000020	0.005024±0.000024
Correlation coefficient ↑	0.995794±0.000109	0.996017±0.000017	0.996026±0.000026
Output SNR/dB↑	21.901±0.094	22.214±0.033	22.198±0.046
Peak amplitude error ↓	0.004930±0.000301	0.004126±0.000153	0.004172±0.000145
Absolute peak shift ↓	0.944±0.043	0.963±0.027	0.954±0.032

Replacing the time-domain input with SWT-MRA components reduced MAE from 0.003803 to 0.003684 and RMSE from 0.005200 to 0.005019, while increasing output SNR from 21.901 to 22.214 dB. Peak amplitude error decreased from 0.004930 to 0.004126. Relative to the time-domain CNN, this configuration reduced MAE and RMSE by 3.14% and 3.48%, respectively, and increased output SNR by 0.313 dB. Across 45 sensor–temperature groups, MAE improved in all 45 groups, RMSE in 44, and output SNR in all 45. These results identify explicit multiscale representation as the main source of the performance gain.

Adding component-level supervision produced an MAE of 0.003685, an RMSE of 0.005024, and an output SNR of 22.198 dB, values close to those obtained without component supervision. The no-component-supervision model performed slightly better in MAE, RMSE, output SNR, and peak amplitude error, while the full model achieved slightly better correlation and peak localization.

Across the 45 groups, adding component supervision reduced MAE in 20 groups, reduced RMSE in 16, and increased output SNR in 18. Thus, no consistent additional improvement was observed. Component-level supervision is best interpreted as an auxiliary training constraint rather than the principal source of the observed gain. Because it is absent from inference, the full and no-component-supervision models have the same architecture and computational complexity at test time.

5.4. Results Across Sensors and Temperatures

Performance was also summarized across sensors and temperature conditions using test results from all five random seeds.

As shown in Table 5, the full model achieved lower RMSE than the time-domain CNN for all three sensors. The reductions for sensors 1, 2, and 3 were 3.81%, 2.50%, and 3.79%, respectively, with output-SNR gains of 0.341, 0.220, and 0.330 dB. Peak amplitude error also decreased for all

sensors, with the largest relative improvement for sensor 3. The magnitude of these gains nevertheless varied with the signal characteristics.

Table 5. Comparison across the three sensors: full SWT-MRA-CNN versus time-domain CNN

Sensor	CNN RMSE	Full model RMSE	RMSE reduction ratio	Output SNR increase/dB	Peak amplitude error Reduction (%)
Sensor 1	0.005199	0.005000	3.81%	0.341	13.67%
Sensor 2	0.004981	0.004857	2.50%	0.220	4.27%
Sensor 3	0.005420	0.005215	3.79%	0.330	27.17%

Across room temperature and the 14 conditions between 300 and 1600 °C, the full model achieved lower mean RMSE than the time-domain CNN, with reductions of approximately 2.5%–4.1%. At 1600 °C, RMSE decreased from 0.006070 to 0.005830 (approximately 3.95%), and output SNR increased from 19.936 to 20.254 dB.

These findings agree with the analysis of the 45 sensor–temperature groups: improvements were observed across the evaluated conditions rather than being confined to a particular sensor or temperature range. Because amplitude, noise, and echo shape can vary with temperature, the results indicate consistency in the direction of improvement, not a temperature-independent gain.

5.5. Qualitative Waveform Comparison

Nine samples were selected by a predetermined rule: the third repetition at room temperature, 900 °C, and 1600 °C for each of the three sensors. Learning-based outputs were obtained with the seed-42 checkpoints, and SWT soft thresholding used the Universal rule selected on the validation set. To keep the visualization readable, Figure 3 displays the sensor-2 waveform at 1600 °C. Panels (a) and (b) show the full waveform and an enlarged view of the principal echo, while panel (c) compares the local absolute errors of the four denoising methods relative to the reference. SWT soft thresholding suppresses some high-frequency noise but exhibits amplitude attenuation near the echo. The time-domain CNN and 1D U-Net reduce local disturbances, although discrepancies remain at some peaks and troughs. SWT-MRA-CNN follows the reference waveform more closely in the displayed echo region and generally shows smaller local errors.

Among the nine selected samples, the full SWT-MRA-CNN achieved the lowest RMSE and highest correlation coefficient for eight. Its mean RMSE was 0.005035, compared with 0.005163 for the time-domain CNN, 0.005604 for 1D U-Net, and 0.007499 for SWT soft thresholding. These findings are consistent with the overall results from 225 test waveforms.

The full model was not the best method at every sample or time point. At low noise levels, the output waveforms were often similar across models. The qualitative plots are therefore used to illustrate waveform characteristics, while the quantitative evaluation over the complete test set remains the primary basis for comparison.

The full model and its no-component-supervision counterpart also produced nearly identical waveforms, with a mean RMSE difference of approximately 0.000007 across the nine selected samples. This agrees with the ablation findings that the multiscale representation accounts for most of the gain, while auxiliary component supervision contributes little additional waveform improvement.

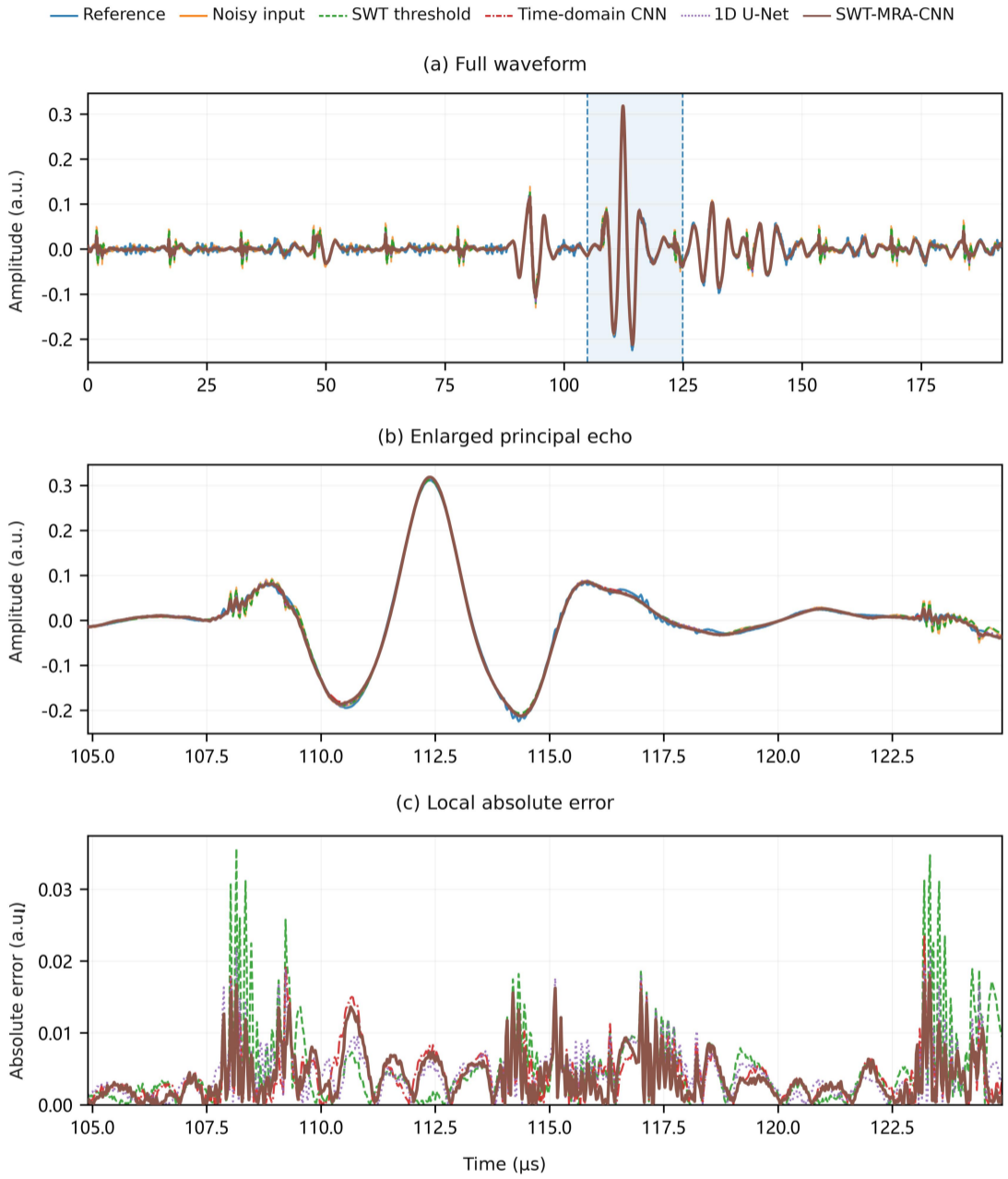


Figure 3. Waveform recovery for sensor 2 at 1600 °C: (a) complete waveform, (b) enlarged principal echo, and (c) local absolute error.

5.6. Model Complexity and Inference Efficiency

5.6.1. Model Size

Table 6 reports the parameter count, MACs, checkpoint size, and peak GPU memory usage for each neural network.

Table 6. Neural-network model complexity

Method	Parameters	Conv/Linear GMACs	Weight-file size (MiB)	Peak GPU memory (B=1, MiB)
Time-domain CNN	261,073	1.9870	1.0149	8.134
1D U-Net	302,017	0.2830	1.1778	3.642
MRA-CNN (no component loss)	266,457	2.0283	1.0354	8.653
Full SWT-MRA-CNN	266,457	2.0283	1.0354	8.653

The full SWT-MRA-CNN contains 266,457 trainable parameters, 2.06% more than the time-domain CNN. Conv1d/Linear MACs, checkpoint size, and peak GPU memory at batch size 1 increase by 2.08%, 2.02%, and 6.38%, respectively. The model's improved denoising accuracy therefore does not require a large increase in CNN parameter count.

The full model has approximately 11.77% fewer parameters than 1D U-Net while attaining lower MAE and RMSE. MACs in Table 6 include only Conv1d and Linear layers; they exclude SWT-MRA construction and elementwise operations. They should therefore be interpreted as estimates of neural-network computation rather than end-to-end processing cost.

5.6.2. GPU-Only Forward-Pass Efficiency

Table 7 reports GPU-only network inference efficiency. At batch size 1, the full SWT-MRA-CNN had a mean forward-pass latency of 4.256 ms, 7.84% higher than the time-domain CNN but 21.36% lower than 1D U-Net. Thus, when SWT-MRA construction is excluded, the nine-channel input causes only a modest increase in single-sample CNN inference time.

Table 7. GPU-only forward-pass efficiency

Method	B = 1 (mean/ms)	B=1 P95/ms	B = 1. Throughput rate / sample·s $^{-1}$	B = 8 mean/ms	B = 8 throughput/sample·s $^{-1}$
Time-domain CNN	3.946	4.324	253.39	12.488	640.59
1D U-Net	5.412	6.853	184.79	5.184	1543.35
MRA-CNN No component loss	4.183	4.698	239.04	13.400	597.01
Full SWT-MRA-CNN	4.256	4.764	234.97	13.440	595.22

At batch size 8, 1D U-Net achieved the highest throughput, although its batch-size-1 latency exceeded that of the full SWT-MRA-CNN. This illustrates that measured GPU performance depends on the architecture's parallel execution characteristics and is not determined by theoretical MACs alone.

5.6.3. End-to-End Inference Efficiency

Table 8 summarizes end-to-end inference efficiency. Unlike GPU-only forward-pass timing, these measurements include SWT-MRA preprocessing where applicable, data transfer, network evaluation, and waveform reconstruction.

Table 8. End-to-end inference efficiency

Method	Average latency/ms	P95 latency/ms	Throughput/sample·s $^{-1}$	Total time for 225 samples / s
Time-domain CNN	4.213	5.445	237.37	0.948
1D U-Net	5.520	7.102	181.17	1.242
MRA-CNN No component loss	57.576	60.540	17.37	12.955
Full SWT-MRA-CNN	58.137	60.290	17.20	13.081
SWT Soft Threshold (CPU)	6.299	6.725	158.76	1.417

The full SWT-MRA-CNN required 58.137 ms per sample on average, compared with 4.213 ms for the time-domain CNN and 5.520 ms for 1D U-Net. Its GPU-only forward pass required 4.256 ms, indicating that most of the additional end-to-end time arose outside the CNN itself.

An average of 38.358 ms was spent constructing the SWT-MRA representation, accounting for 65.98% of total latency. CPU-to-GPU transfer contributed 0.96%, network inference and reconstruction 32.84%, and GPU-to-CPU transfer 0.22%. CPU-side SWT-MRA preprocessing was therefore the main efficiency bottleneck in the present implementation.

SWT soft thresholding required 6.299 ms per sample end to end, but it was implemented entirely on the CPU. Its latency should not be interpreted as a direct same-device hardware comparison with GPU-based methods. Overall, the explicit SWT-MRA representation improved denoising accuracy with only a small increase in network size, but the CPU implementation substantially increased total latency. Potential deployment optimizations include GPU-parallel SWT, faster filter banks, and batched MRA construction.

5.7. Limitations

Several limitations remain. First, the leave-one-out averaged waveform is a pseudo-clean reference rather than physically noise-free ground truth. The reported metrics therefore quantify agreement with this reference, not recovery against an independently measured noise-free signal.

Second, although experimental batches were separated between training, validation, and testing, the same three sensors were present in each batch. Generalization to previously unseen sensors has not been established. Finally, CPU-based SWT-MRA construction imposes substantial end-to-end latency. Future work should examine improved reference construction, unseen-sensor evaluation, and faster wavelet preprocessing.

6. Conclusion

This study developed an ultrasonic signal denoising method combining SWT-MRA with a one-dimensional CNN. SWT-MRA converts each waveform into equal-length multiscale components, and a dilated residual network jointly estimates denoised components while preserving temporal resolution.

Across five random seeds on the independent test batch, the method achieved an MAE of 0.003685, an RMSE of 0.005024, and an output SNR of 22.198 dB. Relative to the time-domain CNN, MAE and RMSE decreased by 3.10% and 3.39%, and output SNR increased by 0.297 dB. Relative to 1D U-Net, MAE and RMSE decreased by 9.22% and 10.19%. Improvements were observed across the evaluated sensor and temperature conditions. Ablation analysis showed that explicit SWT-MRA representation accounted for most of the gain; component-level auxiliary supervision provided only limited additional benefit.

With 266,457 trainable parameters, the full model was only 2.06% larger than the time-domain CNN. However, end-to-end inference required 58.137 ms per sample, of which SWT-MRA construction accounted for 65.98%. Thus, the present CPU-based preprocessing is the main barrier to low-latency deployment. Future work will investigate improved reference signals, generalization to unseen sensors and broader acquisition conditions, and accelerated or learnable multiscale decomposition.

Author Contributions:

Conceptualization, W.J.Q. and Y.H.W.; methodology, W.J.Q.; software, Y.H.W.; validation, W.J.Q., Y.H.W. and G.H.D.; formal analysis, G.H.D.; investigation, W.J.Q.; resources, G.H.D.; data curation, Y.H.W.; writing—original draft preparation, Y.H.W.; writing—review and editing, W.J.Q.; visualization, G.H.D.; supervision, W.J.Q.; project administration, Y.H.W.; funding acquisition, W.J.Q. All authors have read and agreed to the published version of the manuscript.

Funding:

This research received no external funding.

Institutional Review Board Statement:

Not applicable.

Informed Consent Statement:

Not applicable.

Data Availability Statement:

The datasets analyzed during the current study are available from the corresponding author on reasonable request.

Conflict of Interest:

The authors declare no conflict of interest.

References

Abbate, A., Koay, J., Frankel, J., et al. (1997). Signal detection and noise suppression using a wavelet transform signal processor: Application to ultrasonic flaw detection. IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, 44(1), 14-26.

Cantero-Chinchilla, S., Wilcox, P. D., & Croxford, A. J. (2022). Deep learning in automated ultrasonic NDE—Developments, axioms and opportunities. NDT & E International, 131, 102703.

Cui, X., Li, D., Li, Z., et al. (2022). A GAN noise modeling based blind denoising method for guided waves. Measurement, 188, 110596.

Donoho, D. L. (1995). De-noising by soft-thresholding. IEEE Transactions on Information Theory, 41(3), 613-627.

Jadhav, S., Kuchibhotla, R., Agarwal, K., et al. (2023). Deep learning-based denoising of acoustic images generated with point contact method. Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems, 6(3), 031002.

Lázaro, J. C., San Emeterio, J. L., Ramos, A., et al. (2002). Influence of thresholding procedures in ultrasonic grain noise reduction using wavelets. Ultrasonics, 40(1-8), 263-267.

Li, X., Wang, Y., Wan, X., et al. (2023). Physics-informed deep filtering of ultrasonic guided waves for incipient defect inspection of large-scale square tube structures. Journal of Sound and Vibration, 567, 118066.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. International Conference on Learning Representations.

Mallat, S. G. (1989). A theory for multiresolution signal decomposition: The wavelet representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7), 674-693.

Nason, G. P., & Silverman, B. W. (1995). The stationary wavelet transform and some statistical applications. In A. Antoniadis & G. Oppenheim (Eds.), Wavelets and Statistics (pp. 281-300). Springer.

Pardo, E., San Emeterio, J. L., Rodriguez, M. A., et al. (2006). Noise reduction in ultrasonic NDT using undecimated wavelet transforms. Ultrasonics, 44(Suppl), e1063-e1067.

Qian, C., Xia, S., Zhang, P., et al. (2026). Application-specific guided-wave ultrasonic signal denoising: Knowledge-guided synthetic data pipeline and wavelet-initialized attention U-Net. Ultrasonics, 164, 108054.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015 (pp. 234-241). Springer.

Schmidt, P. L., Walker, D. G., Yuhas, D. J., et al. (2014). Thermal measurements using ultrasonic acoustical pyrometry. Ultrasonics, 54(4), 1029-1036.

Singh, H., Ahmed, A. S., Melandsø, F., et al. (2023). Ultrasonic image denoising using machine learning in point contact excitation and detection method. Ultrasonics, 127, 106834.

Tsai, W. Y., Chen, H. C., & Liao, T. L. (2006). High accuracy ultrasonic air temperature measurement using multi-frequency continuous wave. Sensors and Actuators A: Physical, 132(2), 526-532.

Wu, J., Yang, Y., Lin, Z., et al. (2025). Weak ultrasonic guided wave signal recognition based on one-dimensional convolutional neural network denoising autoencoder and its application to small defect detection in pipelines. Measurement, 242, 116234.

Zhang, Z., Du, X., Gao, F., et al. (2026). An improved U-Net method for denoising ultrasonic echo signals in carbon fiber composites. Ultrasonics, 157, 107782.

License: Copyright (c) 2026 Author.
All articles published in this journal are licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0). This license permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited. Authors retain copyright of their work, and readers are free to copy, share, adapt, and build upon the material for any purpose, including commercial use, as long as appropriate attribution is given.