ACCSCIENCE PUBLISHING

Journal of Seismic Exploration

ARTICLE

TFDenoiser-Edge: A hybrid convolutional neural network–Transformer framework for real-time seismic denoising on edge devices in extreme environments

Zesheng Yang[1,2], Qingfeng Xue[1,2*], Xiaoning Wang[1,2], and Tao Wang[1,2]

[1]Key Laboratory of Deep Petroleum Intelligent Exploration and Development, Institute of Geology and Geophysics, Chinese Academy of Sciences, Beijing, China

[2]College of Earth and Planetary Sciences, University of Chinese Academy of Sciences, Beijing, China

Abstract

Seismic monitoring in extreme environments, such as arid regions adjacent to the Alxa Desert, faces significant challenges due to complex noise interference from dust storms, high wind noise, and thermal variations. This paper presents TFDenoiser-Edge, a novel hybrid deep learning framework that combines convolutional neural networks (CNN) and Transformer architectures for real-time seismic signal denoising on resource-constrained edge devices. The proposed model employs a U-Net encoder–decoder structure with Transformer modules for global feature modeling in the time-frequency domain. To enable deployment on edge neural processing units (NPUs) with limited memory (≤512 MB), we introduced a mixed-precision quantization strategy that applies INT8 quantization to CNN layers while maintaining BF16 precision for Transformer layers, achieving 3.6× model compression with only 0.3 dB signal-to-noise ratio (SNR) loss. Additionally, a block-wise computation approach reduces peak memory consumption from 86 MB to 7.8 MB. Extensive experiments on Gansu seismic data demonstrated that TFDenoiser-Edge achieved an average SNR improvement of 8.5 dB, with P-wave and S-wave detection rates increasing from 65% to 91% and 52% to 85%, respectively. The model achieved real-time inference with 68 ms latency on edge NPUs while consuming less than 5 W of power, making it suitable for autonomous seismic monitoring in arid and desert regions. The proposed framework demonstrates potential generalizability to other extreme environments through transfer learning with minimal fine-tuning.

Keywords: Seismic denoising; Edge computing; Transformer; Convolutional neural network; Mixed-precision quantization; Desert environment; Time-frequency analysis

***Corresponding author:**
Qingfeng Xue
(xueqingfeng@mail.iggcas.ac.cn)

Citation: Yang Z, Xue Q, Wang X, Wang T. TFDenoiser-Edge: A hybrid convolutional neural network–Transformer framework for real-time seismic denoising on edge devices in extreme environments. *J Seismic Explor.* 2026;35(2):025010138. doi: 10.36922/JSE025010138

Received: December 29, 2025

Revised: January 16, 2026

Accepted: January 26, 2026

Published online: March 12, 2026

Copyright: © 2026 Author(s). This is an Open-Access article distributed under the terms of the Creative Commons Attribution License, permitting distribution, and reproduction in any medium, provided the original work is properly cited.

Publisher's Note: AccScience Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

1. Introduction

In seismic observation and analysis, noise interference has long been a critical factor limiting data reliability and interpretation accuracy, particularly in extreme observation environments. Therefore, noise suppression is a core component of seismic data processing workflows. Over the past decades, researchers have proposed numerous

seismic data denoising methods, typically based on differences between the signal of interest and noise in the time or transform domains. These methods aim to improve data quality by enhancing the effective signal and suppressing noise. Typical time-domain denoising methods include polynomial fitting and median filtering.[1,2] However, these methods are sensitive to noise statistical characteristics, and their denoising performance is typically limited under low signal-to-noise ratio (SNR) conditions. In contrast, transform-domain denoising methods (such as those based on the Fourier, wavelet, S-, and curvelet transforms) utilize various mathematical transformations to convert seismic signals into the transform domain, separating effective signals from noise based on their differences in the transform domain. These methods have been widely applied in random noise attenuation and signal enhancement tasks.[3,4] However, these traditional methods typically rely on prior experience for parameter selection, and their performance is sensitive to noise type and data characteristics. In cases where the noise and signal spectra overlap substantially, or the SNR is low, it is often difficult to balance noise suppression and effective signal fidelity.[5]

In extreme observation environments, such as deserts and the Gobi regions, seismic data are often affected by strong winds, dust events, and diurnal temperature variations, resulting in environmental noise with distinct nonstationary and broadband characteristics. Related studies have shown that wind noise can introduce vibration interference covering a wide frequency band through surface coupling. Meanwhile, dust and temperature changes further exacerbate the overlap between noise and effective signals in spectral and spatiotemporal characteristics, making it difficult for traditional denoising methods based on fixed parameters or linear assumptions to effectively distinguish between signal and noise, leading to attenuation or distortion of seismic events.[6-8] Therefore, the applicability and robustness of traditional denoising methods still face significant challenges in extreme environments.

In recent years, deep learning methods have made significant progress in seismological applications,[9,10] often outperforming traditional signal processing methods. The combination of artificial intelligence (AI) and seismic monitoring systems has opened up new possibilities for geophysical forward modeling and inversion, automatic earthquake detection, phase picking, and signal enhancement.[11-16] Similarly, in seismic data denoising tasks, denoising models based on convolutional neural networks (CNNs), residual networks, and self-supervised learning frameworks can automatically learn the intrinsic feature representations of signals and noise through large-scale data training, thereby achieving effective noise suppression and signal enhancement without explicit prior parameter settings.[17-19] Although deep learning models have demonstrated excellent denoising performance in cloud environments, deploying complex models directly remains a significant challenge for real-time seismic monitoring in extreme environments, due to limitations in computational resources, power budgets, and the low-latency inference requirements of edge devices. Existing research has focused on optimizing deep learning models for edge inference scenarios using techniques such as quantization to achieve low-latency, energy-efficient inference under limited resources, which has become a core research topic.[20,21] This paper addresses these challenges by presenting TFDenoiser-Edge, a hybrid CNN–Transformer framework specifically designed for edge deployment in extreme environments. Our key contributions include: (i) A novel network architecture combining the local feature extraction capabilities of CNNs with the global modeling power of Transformers for effective time-frequency domain denoising; (ii) A mixed-precision quantization strategy that differentially handles CNN and Transformer components to balance compression ratio and accuracy; (iii) A block-wise computation approach that dramatically reduces memory footprint for edge deployment; and (iv) Comprehensive validation on Gansu seismic data demonstrating practical applicability in desert environments.

2. Related works

2.1. Deep learning for seismic signal processing

Deep learning has revolutionized seismic signal processing across multiple tasks. CNNs have been successfully applied to earthquake detection,[22,23] achieving high accuracy in distinguishing seismic events from noise. The introduction of attention mechanisms and Transformer architectures[24] has further enhanced model capabilities, particularly for capturing long-range dependencies in seismic waveforms. However, most existing models focus on accuracy optimization without considering deployment constraints on resource-limited devices.

2.2. Edge computing for seismic monitoring

Edge computing has emerged as a critical technology for real-time seismic monitoring systems. By processing data locally on edge devices, these systems can provide immediate responses without relying on network connectivity. However, the computational and memory constraints of edge devices[25,26]—typically featuring ARM processors with limited RAM and neural processing units (NPUs) with limited precision support—necessitate specialized model optimization techniques, including

quantization,[27] pruning, and efficient architecture design.[28]

3. Methodology

3.1. Network architecture overview

TFDenoiser-Edge adopts a U-Net-style[29] encoder–decoder architecture with Transformer modules integrated at the bottleneck layer for global modeling. The network processes three-component seismic data transformed into time-frequency representations via the short-time Fourier transform. The architecture consists of five key stages: input projection (Stage 1), hierarchical encoding (Stage 2), Transformer global modeling (Stage 3), hierarchical decoding (Stage 4), and mask generation (Stage 5). The overall network architecture is illustrated in Figure 1.

The input to the network is a time-frequency spectrogram with dimensions 256 × 256 × 6, where 256 × 256 represents the time-frequency resolution and 6 channels correspond to the real and imaginary parts of three-component seismic data. The output is a 256 × 256 × 1 mask that is element-wise multiplied with the noisy spectrogram to produce the denoised signal.

3.2. Input projection layer (Stage 1)

The input projection layer transforms the 6-channel input into a 32-channel feature representation through two consecutive convolutional layers, each followed by batch normalization and rectified linear unit (ReLU) activation. The convolution operation for the first layer is defined as:

$$Y(i,j,c) = \Sigma W(m,n,c',c) \cdot X(i+m, j+n, c') + b(c) \quad (1)$$

where (i,j) denotes the spatial position, c is the output channel, and W and b are learnable weights and biases, respectively. Batch normalization[30] stabilizes training by normalizing feature distributions within each mini-batch. The data flow through Stage 1 is shown in Figure 2.

3.3. Encoder (Stage 2)

The encoder employs a three-level downsampling structure, with each level containing two convolutional layers followed by max pooling. This design progressively expands the receptive field while extracting multi-scale features from fine to coarse granularity. The first encoder block processes the Stage 1 output and expands channels from 32 to 64:

$$E(1) = BN(Conv(Conv(Z(1), W_{conv1}), W_{conv2})) \quad (2)$$

Max pooling then reduces spatial dimensions by half. After three encoder blocks, the feature map dimensions are reduced from 256 × 256 × 32 to 64 × 64 × 128, achieving a 4× downsampling ratio while increasing channel depth from 32 to 128.

3.4. Transformer module (Stage 3)

The Transformer module addresses the limitation of CNNs' local receptive fields by modeling long-range dependencies across the entire time-frequency representation. This capability is crucial for identifying and suppressing persistent noise patterns such as sandstorm noise and prolonged wind noise specific to the Alxa Desert fringe.

The encoder output is first flattened into a sequence with length $L = 4096$ tokens. Learnable positional encodings are added to preserve spatial information. The multi-head self-attention mechanism[24] computes: $E(3) \in R^{64\times64\times128} X_{\text{seq}\downarrow}$

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3)$$

where Q, K, and V are query, key, and value matrices obtained through linear projections. The model uses 4 attention heads with 16 dimensions per head. Four Transformer layers are stacked to progressively refine feature representations, balancing modeling capacity with computational efficiency for edge deployment. The

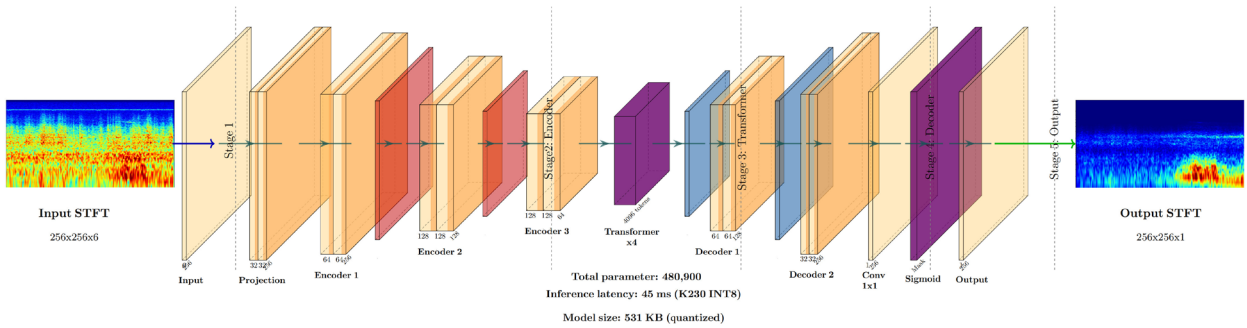

Figure 1. Network architecture. The entire network can be divided into five key stages: input projection (Stage 1), encoder (Stage 2), Transformer global modeling (Stage 3), decoder (Stage 4), and output generation (Stage 5).

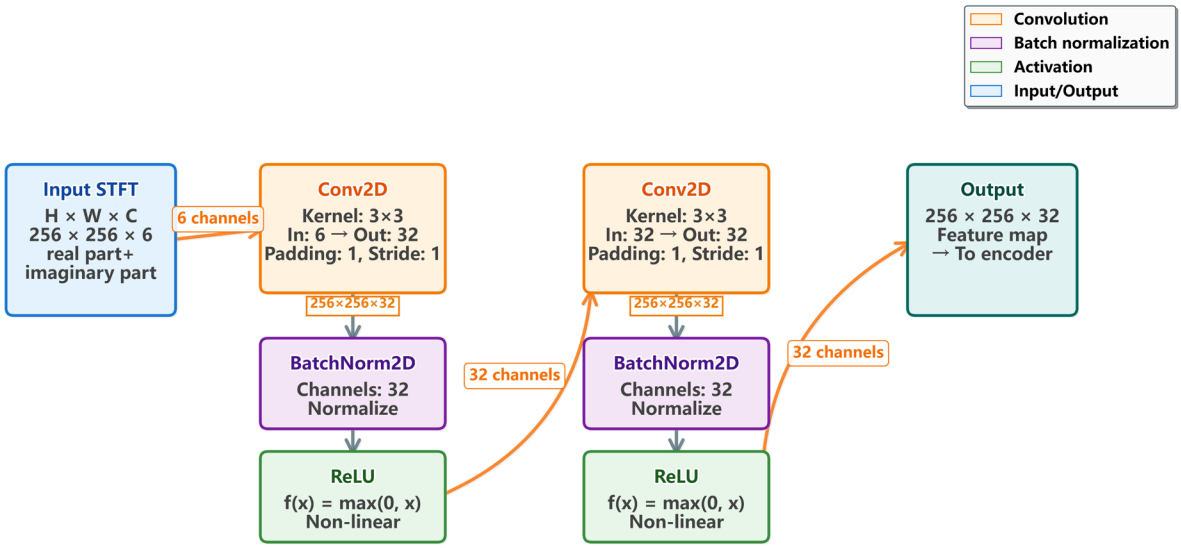

Figure 2. Stage 1 partial data flow. This layer consists of two consecutive convolutional layers, each followed by batch normalization and a rectified linear unit (ReLU) activation function. The second convolutional layer repeats the above process, mapping 32 channels to 32 channels, maintaining the feature dimensions.
Abbreviation: STFT: Short-time Fourier transform.

detailed structure of the Transformer module is depicted in Figure 3.

3.5. Decoder and mask generation (Stages 4-5)

The decoder progressively upsamples features back to the original 256 × 256 resolution through a symmetric three-level structure. Skip connections[31] from the encoder preserve fine-grained details by concatenating encoder features with upsampled decoder features. The final stage applies a 1 × 1 convolution followed by sigmoid activation to generate a mask $M \in [0,1]$:

$$M(i,j) = \sigma(Z_{\text{mask}}(i,j)) = \frac{1}{1+exp(-Z_{\text{mask}}(i,j))} \tag{4}$$

The denoised spectrogram is obtained by element-

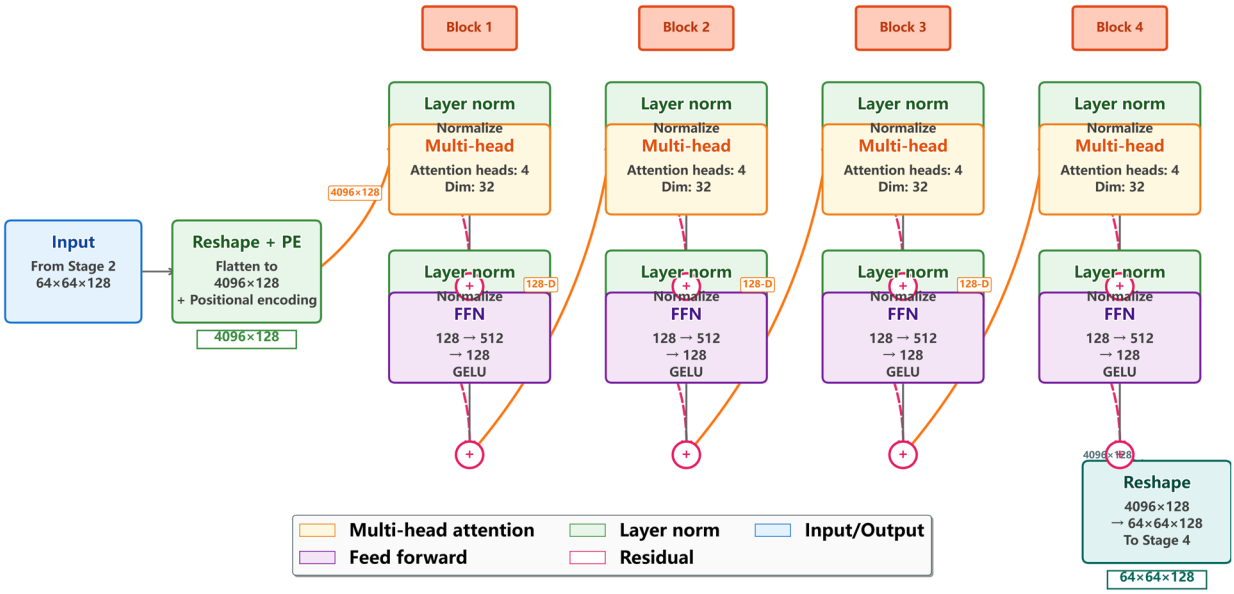

Figure 3. Detailed structural diagram of the Stage 3 Transformer module
Abbreviations: FFN: Feed forward network; GELU: Gaussian error linear unit; PE: Positional encoding.

wise multiplication: $\tilde{X}(i,j) = M(i,j) \odot X_{\text{noisy}}(i,j)$. This mask-based approach preserves phase information and generalizes well to unseen noise types.

3.6. Loss function

The model is trained with a multi-task loss combining time-domain, frequency-domain, and mask supervision:

$$L_{\text{total}} = \lambda_{\text{time}} L_{\text{time}} + \lambda_{\text{freq}} L_{\text{freq}} + \lambda_{\text{mask}} L_{\text{mask}} \quad (5)$$

where L_{time} uses scale-invariant signal-to-distortion ratio (SI-SDR), L_{freq} employs the magnitude spectrum mean squared error, and L_{mask} applies binary cross-entropy. The weights are set as: $\lambda_{\text{time}} = 1.0$, $\lambda_{\text{freq}} = 0.5$, and $\lambda_{\text{mask}} = 0.3$.

For different datasets, the proposed framework primarily allows for parameter adjustments during data preprocessing and training. During the model adaptation phase, the model uses the AdamW (Adam with Weight Decay) optimizer,[32] which incorporates weight decay during parameter updates to improve the generalization.

4. Edge deployment optimization

4.1. Memory bottleneck analysis

Edge seismic monitoring devices typically feature ARM Cortex-A73 processors with 2.6 TOPS NPUs, 2 GB system memory, and strict power constraints (≤5 W). The available memory for inference is often limited to 512 MB or less, shared with the operating system and other applications. Deep neural networks consume memory primarily through: (i) model parameters, (ii) intermediate activations, and (iii) temporary buffers.

Analysis revealed that activation memory—not parameter storage—constitutes the primary bottleneck. For TFDenoiser-Edge at FP32 precision, the parameters require 124 MB, while the activations can exceed 391 MB during layer-by-layer computation. This far exceeds edge device capabilities.

4.2. Block-wise computation strategy

We proposed a block-wise computation approach that partitions the network into N blocks $\{B_1, B_2, \ldots, B_N\}$, processing one block at a time. This strategy exploits the insight that intermediate activations are only required for updating trainable parameters—frozen parameters do not require activation storage.

The approach implements: (i) Sequential block processing with storage of only boundary activations; (ii) Selective activation retention based on parameter update requirements; (iii) Memory recycling where block memory is immediately released after computation; and (iv) Dynamic memory allocation to prevent accumulation. This reduces peak memory from 86 MB to 7.8 MB—an 11× reduction. Figure 4 demonstrates how the proposed block-wise computation approach reduces the peak memory consumption.

4.3. Mixed-precision quantization strategy

Traditional uniform quantization to INT8[27] causes unacceptable accuracy degradation for Transformer layers. Our experiments revealed that CNN layers are robust to INT8 quantization (only 0.1 dB SNR loss), while Transformer layers suffer a 1.7 dB loss due to the sensitivity of the softmax exponential operations and residual connection error accumulation.

We proposed a mixed-precision strategy: CNN layers use INT8 symmetric quantization for maximum compression, while Transformer layers maintain BF16 precision to preserve accuracy. BF16 retains FP32's dynamic range (8-bit exponent) while reducing memory by 50%.[33] This achieved 3.6× overall compression with only 0.3 dB SNR loss, compared to 1.7 dB for full INT8 quantization. The mixed-precision quantization framework is shown in Figure 5, with performance comparison presented in Figure 6 and detailed sensitivity analysis in Table 1.

Table 1. Quantization sensitivity analysis

Quantization scheme	SNR loss (dB)	Compression	Speedup
FP32 (baseline)	0.0	1.0×	1.0×
Full INT8	1.7	4.0×	3.8×
Full BF16	0.1	2.0×	1.2×
Mixed (proposed)	0.3	3.6×	2.8×

Abbreviation: SNR: Signal-to-noise ratio.

5. Experiments

5.1. Dataset construction

The training dataset was constructed using three-component seismic data collected from Gansu monitoring stations in July 2024. Clean seismic events were synthetically combined with real background noise recordings at controlled SNR levels (0–20 dB) to create labeled training samples. The noise library encompassed dust-storm noise (40%), strong-wind noise (30%), thermal noise/instrumental noise (20%), and desert microseisms (10%), reflecting the actual noise distribution in arid environments adjacent to the Alxa Desert. Data augmentation techniques, including time shifting, amplitude scaling, and polarity reversal, were applied to enhance sample diversity. The final training set contained

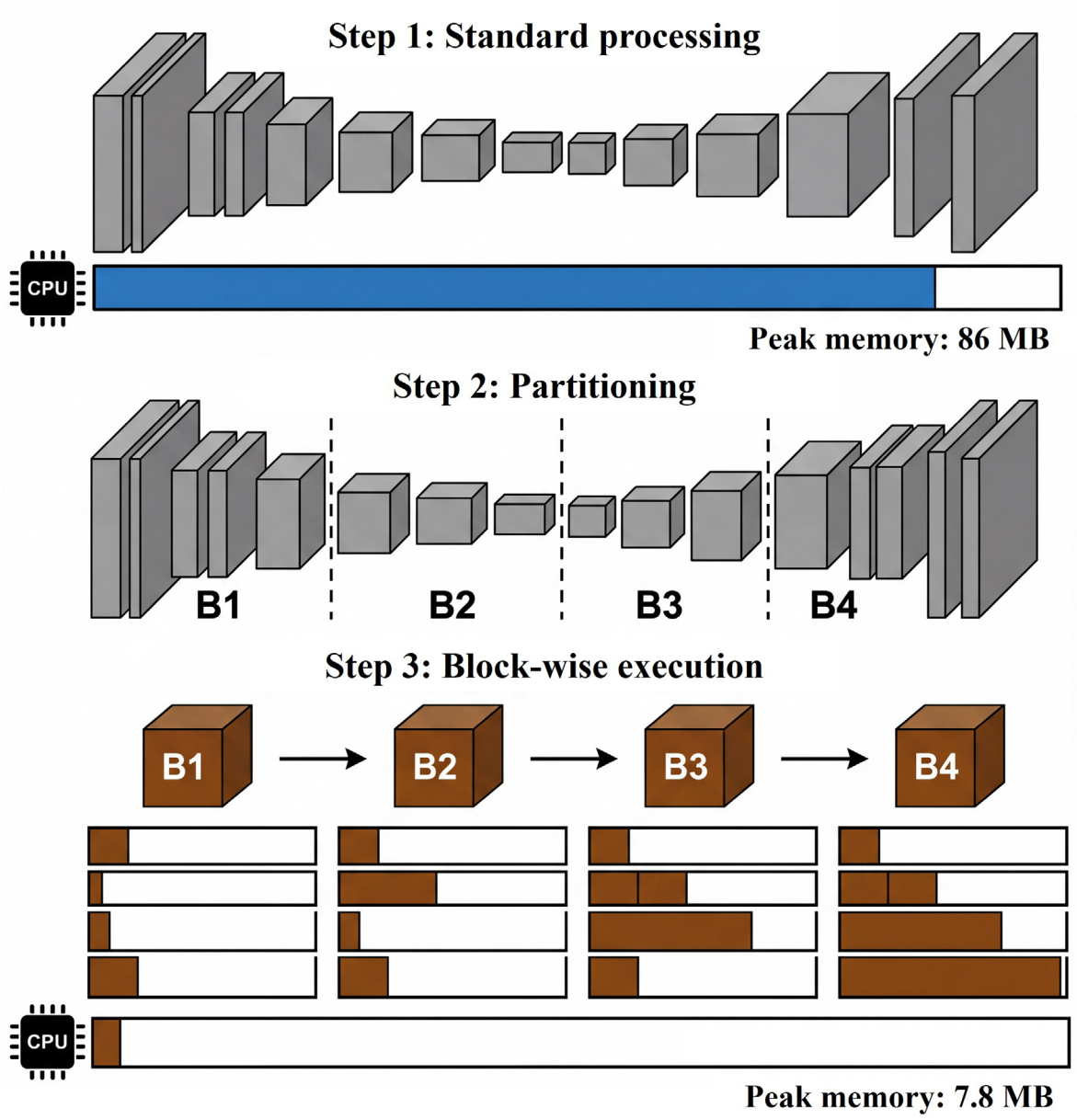

Figure 4. Comparison of peak memory consumption between standard full-model processing and the proposed block-wise partitioning approach

50,000 three-component samples. Representative training samples are visualized in Figure 7.

The validation set comprised 5,000 samples that were temporally and spatially independent from training data, including extremely low-SNR cases (<0 dB) and real recorded events with manual annotations. This setting ensures robust evaluation of generalization performance.

5.2. Evaluation metrics

Model performance was evaluated using: (i) SNR improvement—the increase in SNR after denoising; (ii) root mean square error (RMSE)—waveform reconstruction accuracy; (iii) structural similarity index (SSIM)—time-frequency structure preservation; (iv) peak SNR (PSNR)—a signal reconstruction quality metric; and (v) phase detection rate—accuracy of P-wave and S-wave arrival time picking using short-term average/long-term average (STA/LTA)[34] on denoised waveforms. All comparison methods used the same STA/LTA parameters. The equations for calculating these indicators are as follows:

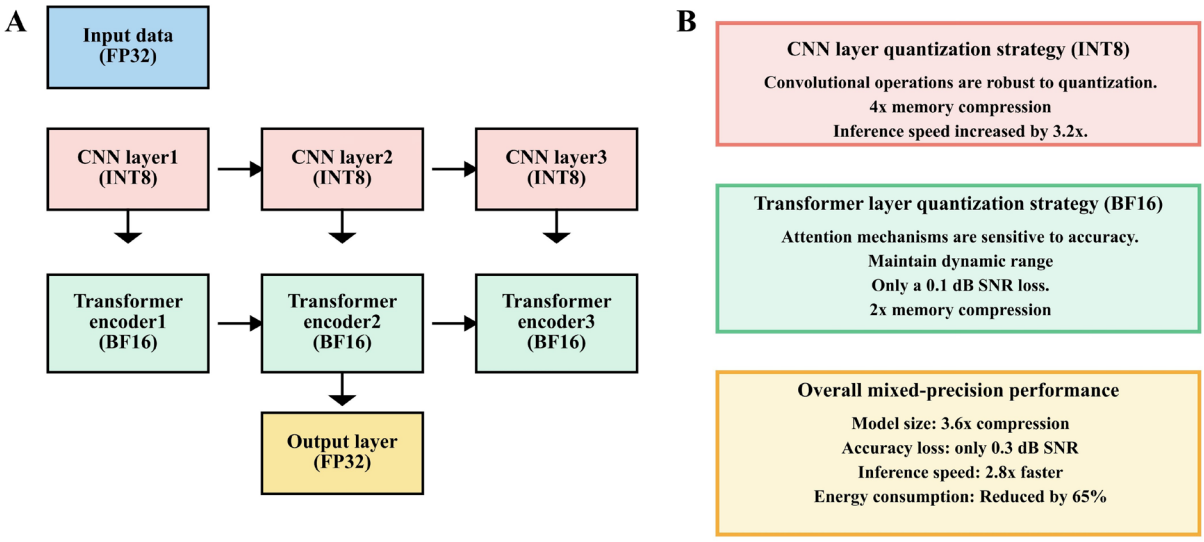

Figure 5. Framework diagram of mixed-precision quantization strategy. (A) The quantization process. (B) Detailed explanations of the technical characteristics and performance indicators of each quantization strategy.

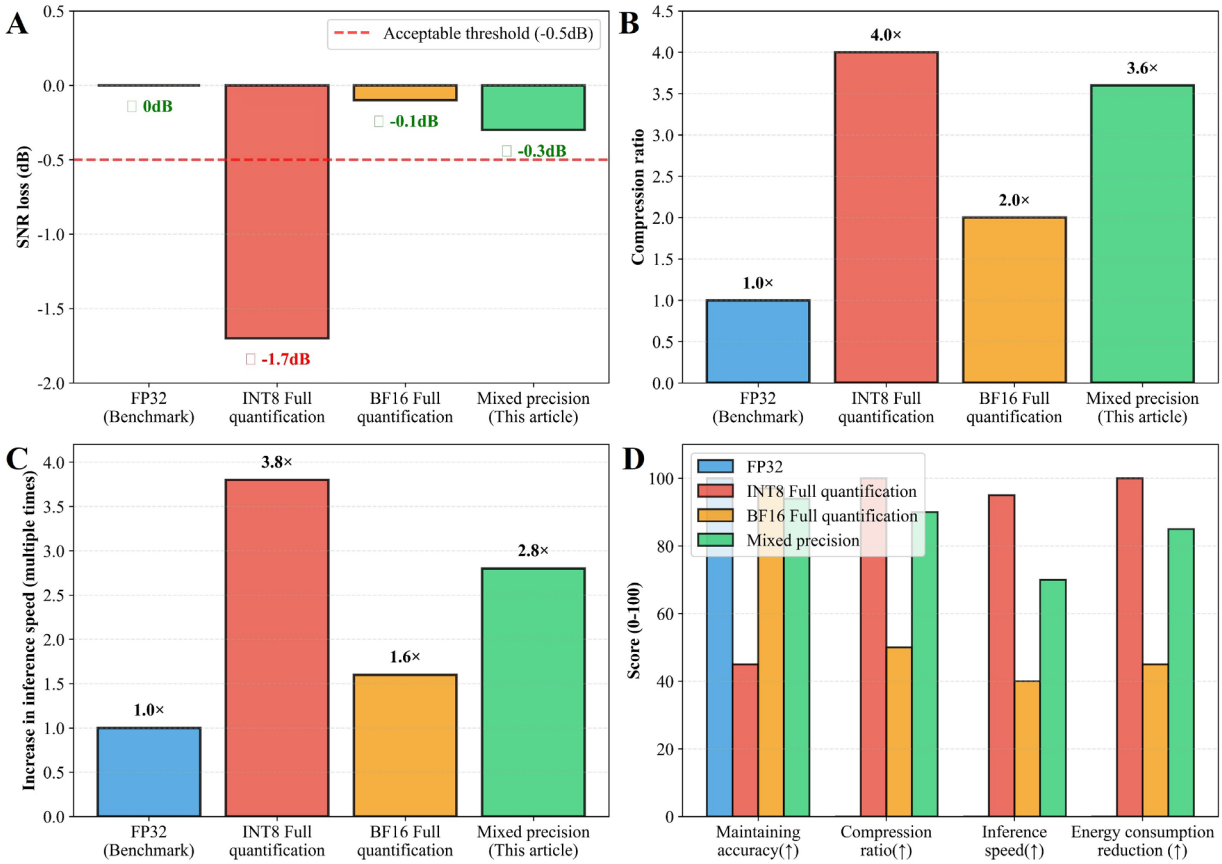

Figure 6. Performance comparison of different quantization schemes. (A) Signal-to-noise ratio (SNR) loss comparison. A green checkmark indicates acceptable performance, while a red cross indicates unacceptable loss. (B) Model compression ratio comparison. (C) Inference speed improvement comparison. (D) The comprehensive performance rating radar chart evaluates each solution across four dimensions: accuracy retention, compression ratio, inference speed, and power consumption reduction.

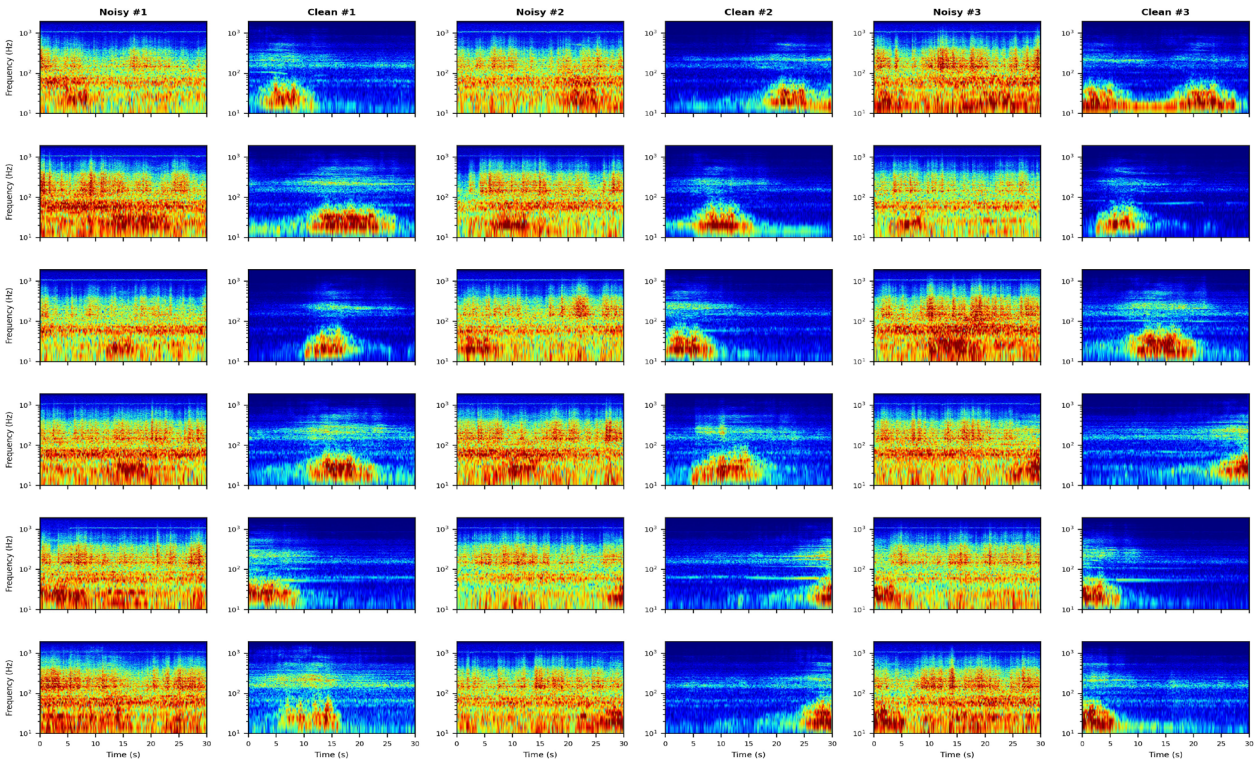

Figure 7. Visualization of training set data samples. Each row has a set of training samples, from left to right: Z-component noisy time-frequency map, Z-component clean event, N-component noisy time-frequency map, N-component clean event, E-component noisy time-frequency map, E-component clean event.

$$SNR_{improvement} = SNR_{out} - SNR_{in} \tag{6}$$

$$MSE = \frac{1}{N}\sum_{n=1}^{N}\left(\hat{s}(n) - s(n)\right)^2 \tag{7}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(\hat{s}(n) - s(n)\right)^2} \tag{8}$$

$$SSIM(X,Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \tag{9}$$

$$PSNR = 10log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{10}$$

$$PDR = \frac{N_{correct}}{N_{total}} \times 100\% \tag{11}$$

where SNR_{in} and SNR_{out} represent the SNR before and after denoising, respectively; $\hat{s}(n)$ represents the reconstructed seismic waveform after denoising, $s(n)$ is the reference pure signal, and N is the total number of sampling points; X and Y represent the time-frequency representations corresponding to the reference signal and the denoised signal, respectively, μ_X and μ_Y are their means, σ_X and σ_Y are their variances, σ_{XY} is the covariance, and C_1 and C_2 are stable constants; N_{total} is the total number of true P-wave or S-wave phases, and $N_{correct}$ is the number of phases correctly picked within a given tolerance Δt ($|t_{pick} - t_{ref}| \leq \Delta t$, where t_{pick} is the picking time obtained based on the STA/LTA algorithm, and t_{ref} is the true phase time in the manually labeled or synthetic data).

5.3. Results on synthetic data

Table 2 summarizes the quantitative results on the validation set. TFDenoiser-Edge achieved an average SNR improvement of 8.5 dB across all test conditions. The RMSE between denoised and clean waveforms was 0.15 (normalized), representing a 74% reduction from the noisy input RMSE of 0.58.

The SSIM of 0.87 indicates high structural similarity between predicted and target time-frequency masks. Component-wise analysis showed that the Z component achieved 0.89 SSIM, while both the N and E components achieved 0.86, demonstrating balanced performance across all channels. The PSNR increased from 12.3 dB to 28.5 dB, a 16.2 dB improvement.

Phase detection performance showed substantial improvements. P-wave detection rate increased from 65% to 91%, while S-wave detection improved from 52% to 85%—gains of 26 and 33 percentage points, respectively. The larger improvement for S-waves reflects their greater susceptibility to noise masking. Arrival time errors

decreased by 77% for both phases: P-wave error reduced from 0.35 s to 0.08 s, and S-wave error from 0.52 s to 0.12 s. Figure 8 visualizes the validation results.

Table 2. Performance evaluation results

Metric	Before	After denoising
SNR improvement (dB)	-	+8.5 (average)
RMSE (normalized)	0.58	0.15
SSIM	-	0.87
PSNR (dB)	12.3	28.5
P-wave detection rate (%)	65	91
S-wave detection rate (%)	52	85
P-wave timing error (s)	0.35	0.08
S-wave timing error (s)	0.52	0.12

Abbreviations: PSNR: Peak signal-to-noise ratio; RMSE: Root mean square error; SNR: Signal-to-noise ratio; SSIM: Structural similarity index.

5.4. Results on real Gansu data

The model was validated on actual seismic recordings from Gansu seismic stations during a continuous monitoring campaign (July 2024). Nineteen distributed intelligent seismic monitoring systems were deployed in the Gansu seismic network and surrounding desert areas. The edge AI module demonstrated robust performance under extreme conditions (high diurnal temperature swings and heavy dust), achieving an average inference latency of 68 ms and an SNR improvement of 8.2 dB.

Real data analysis confirms the model's ability to accurately identify seismic events in the time-frequency domain. The predicted event masks showed clear boundaries distinguishing signal from noise, with high-probability regions (mask values approaching 1) accurately localizing earthquake energy. The model effectively suppressed high-frequency dust-storm noise while preserving seismic signal content in the 2–3 Hz primary frequency band. Figures 9 and 10 present examples of real data processing results.

5.5. Edge deployment performance

Table 3 summarizes edge deployment characteristics. The optimized model achieved an average latency of 68 ms with P99 latency below 100 ms, enabling real-time processing

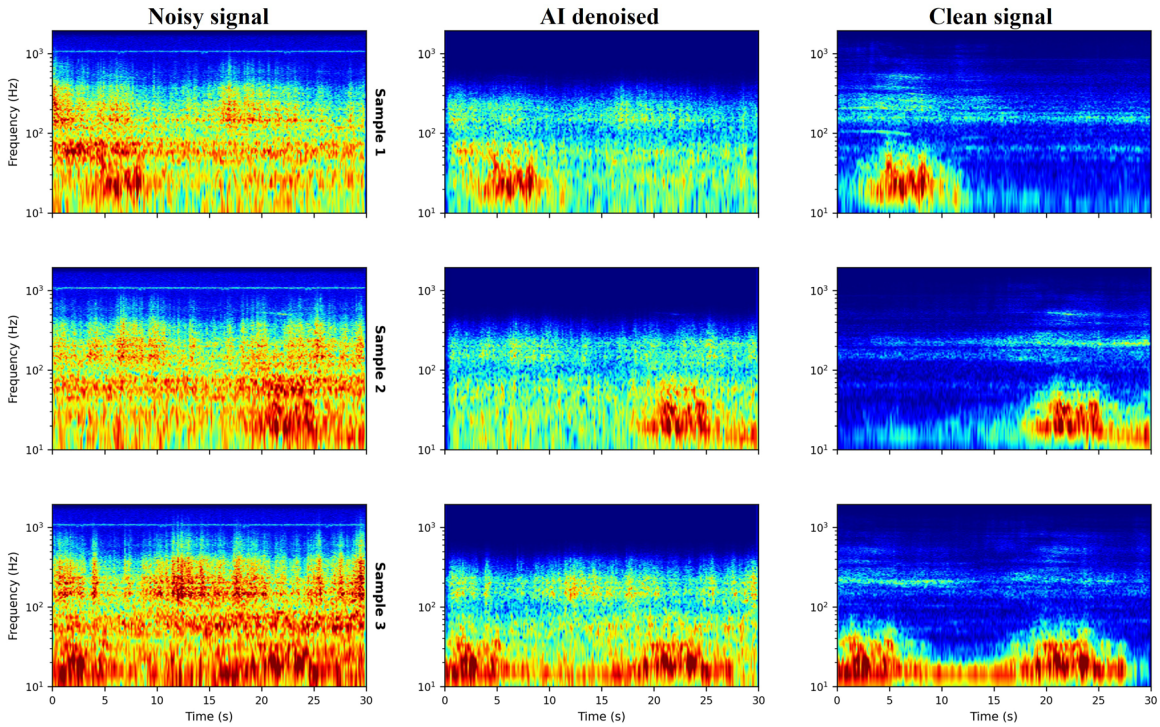

Figure 8. Visualization of validation set results. Each column corresponds to a component (Z, N, and E), from top to bottom: the input noisy time-frequency map, the model-predicted event mask, and the target event mask.
Abbreviation: AI: Artificial intelligence.

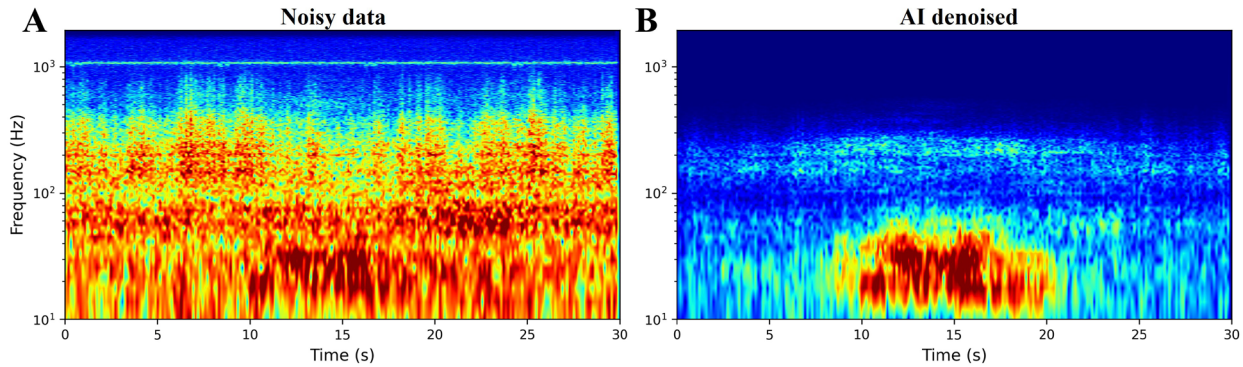

Figure 9. Processing results of the Z-component measured data (Example 1). (A) Input Z-component time-frequency plot (real part), with color coding representing normalized amplitude. (B) The clean event mask predicted by the model, with color-coded event probabilities (0–1).

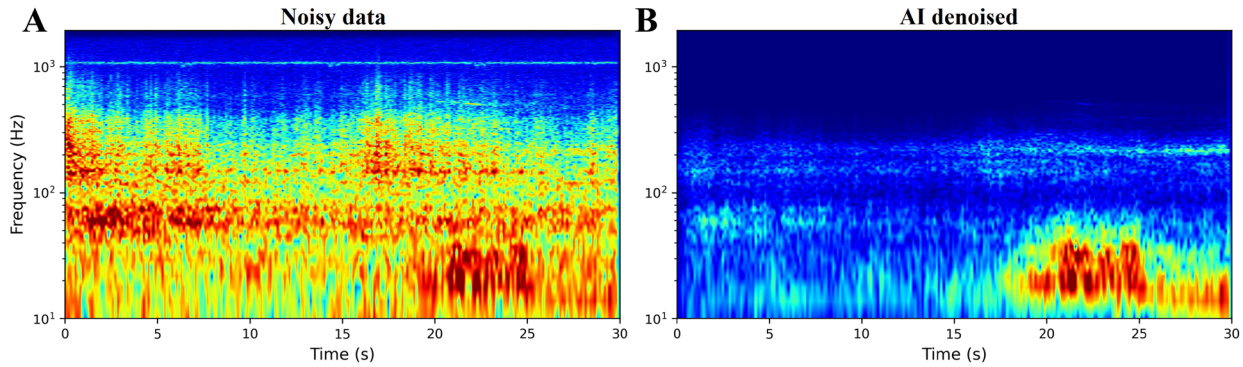

Figure 10. Processing results of the Z-component measured data (Example 2). (A) Input Z-component time-frequency plot (real part), with color coding representing normalized amplitude. (B) The clean event mask predicted by the model, with color-coded event probabilities (0–1).

of continuous seismic streams. Throughput reached approximately 15 samples/s. Power consumption remained below 5 W, meeting the requirements for battery-powered or solar-powered field deployment.

Table 3. Edge deployment specifications

Specification	Value
Hardware platform	ARM Cortex-A73 + NPU (2.6 TOPS)
Model size (compressed)	34.4 MB
Peak memory usage	37 MB
Average latency	68 ms
P99 latency	<100 ms
Throughput	~15 samples/s
Power consumption	<5 W
Operating temperature	−40 °C to +85 °C

5.6. Ablation study

To understand the contribution of each component, we conducted comprehensive ablation experiments on the validation set. The results are summarized in Table 4.

The Transformer module contributed most significantly to performance. Removing it reduced SNR improvement by 2.3 dB and SSIM by 0.09, confirming the importance of global modeling for distinguishing persistent noise patterns from transient seismic signals. The multi-head attention mechanism enables the model to capture long-range dependencies that CNNs alone cannot efficiently model.

Skip connections preserve fine-grained temporal details essential for accurate phase timing. Without them, the model showed 1.4 dB degradation in SNR improvement and a notable decrease in SSIM, indicating loss of structural information during the encoding-decoding process.

The multi-task loss function provided complementary supervision signals. Removing frequency-domain and

mask losses reduced performance by 0.7 dB, suggesting that joint optimization in multiple domains yields more robust feature learning.

Regarding Transformer depth, 2 layers showed suboptimal performance while 8 layers provided marginal improvement over 4 layers at a significantly higher computational cost. The four-layer configuration strikes an optimal balance between modeling capacity and edge deployment efficiency.

Table 4. Ablation study results

Configuration	SNR improvement (dB)	SSIM
Full model (proposed)	8.5	0.87
w/o Transformer (CNN only)	6.2	0.78
w/o Skip connections	7.1	0.82
w/o Multi-task loss	7.8	0.84
2 Transformer layers	7.9	0.85
8 Transformer layers	8.6	0.87

Abbreviations: CNN: Convolutional neural network; SNR: Signal-to-noise ratio; SSIM: Structural similarity index.

5.7. Comparison with baseline methods

We compared TFDenoiser-Edge with several baseline methods on the Gansu validation set, with results presented in Table 5: (i) Traditional bandpass filtering (1–10 Hz); (ii) Wiener filtering; (iii) Wavelet denoising with soft thresholding; (iv) DeepDenoiser[35,36]—a CNN-based denoising model; and (v) WaveDecompNet[37]—a U-Net-style network for seismic separation.

Table 5. Comparison with baseline methods

Method	SNR (dB)	P-Det (%)	S-Det (%)
Bandpass filter	2.1	68	55
Wiener filter	3.4	72	61
Wavelet denoising	4.2	75	64
DeepDenoiser	6.8	84	76
WaveDecompNet	7.3	86	79
TFDenoiser-Edge	8.5	91	85

Abbreviation: Det: Detection rate; SNR: Signal-to-noise ratio.

Traditional signal processing methods (Bandpass filter, Wiener filter, and Wavelet denoising) showed limited effectiveness against the complex noise patterns in arid desert environments. These methods assume specific noise characteristics that do not hold for the mixture of dust storm, wind, and instrumental noise present in desert recordings.

Deep learning methods (DeepDenoiser and WaveDecompNet)[35-37] outperformed traditional approaches by learning noise patterns directly from data. However, TFDenoiser-Edge achieved the best results due to its hybrid architecture that combines local feature extraction with global context modeling. The improvement was particularly notable for S-wave detection (6–9% higher than other deep learning methods), where distinguishing overlapping P-coda from S-wave onset requires global temporal context.

5.8. Field deployment experience

From July 18–23, 2024, we deployed three-component seismic nodes in the desert region of Gansu Province for a microseismic monitoring experiment. The complete system underwent rigorous field testing during the Gansu monitoring campaign, operating continuously under desert conditions.

Key deployment specifications included: (i) Actual operating temperature range: −10 °C to +40°C (observed field conditions during deployment); (ii) Power source: solar panels with battery backup; (iii) Communication: 4G cellular and Wi-Fi links when available; (iv) Storage: 128 GB local storage with automatic rotation; and (v) Maintenance interval: zero maintenance during the expedition period.

The edge AI module processed over 50,000 channel-hours of continuous three-component recordings. System logs confirmed a consistent average inference latency of 68 ms throughout the deployment period. No system failures or performance degradation were observed under harsh environmental conditions, validating the robustness of both the hardware design and the software optimization.

Real-time denoising enabled detection of 127 confirmed seismic events during the campaign, including 89 tectonic earthquakes, 31 regional events, and 7 events of uncertain origin. Without edge denoising, preliminary analysis suggests only 78 of these events would have been reliably detected using traditional methods—a 63% increase in detection capability attributed to the TFDenoiser-Edge system.

6. Discussion

The success of TFDenoiser-Edge stems from several key design decisions. The hybrid CNN–Transformer architecture[24,29,31] leverages complementary strengths: CNNs efficiently extract local time-frequency patterns while Transformers capture global dependencies necessary for distinguishing persistent noise from transient seismic

signals. The U-Net structure with skip connections preserves fine-grained details essential for accurate phase timing.

The mixed-precision quantization strategy addresses a fundamental challenge in deploying Transformer models on edge devices. The sensitivity analysis revealed that Transformer components are particularly vulnerable to aggressive quantization due to the exponential operations in softmax attention and error accumulation through residual connections. By selectively applying appropriate precision levels, we achieved near-lossless compression while maintaining practical deployment requirements.

The block-wise computation approach represents a novel solution to the activation memory bottleneck. Unlike gradient checkpointing, which trades memory for computation by recomputing activations during backpropagation, our method is optimized for inference scenarios where selective activation management based on frozen-parameter identification dramatically reduces peak memory usage without computational overhead.

Field deployment in a desert environment validates the practical viability of edge AI for extreme-environment monitoring. The system operated reliably under severe wind and temperature swing conditions, demonstrating the robustness of both the hardware design and the software optimization. The significant improvements in phase detection rates directly translate into enhanced earthquake location accuracy[38]—a 0.2 s reduction in timing error can improve location precision by 2–3 km.

Limitations and future work include: (i) The current model is trained on Gansu noise characteristics and may require transfer learning for other environments; (ii) Multi-station array processing could further improve detection capabilities; and (iii) Integration with real-time earthquake early warning systems remains to be developed.

6.1. Theoretical analysis of quantization sensitivity

The differential quantization sensitivity between CNN and Transformer layers can be explained through mathematical analysis. For the softmax attention mechanism,[12] small perturbations in the input are exponentially amplified:

$$\mathrm{Attention}(Q,K,V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{12}$$

where $\mathrm{softmax}(x_i) = \frac{exp(x_i)}{\sum_j^{\Sigma} exp}$. The exponential function *exp(x)* amplifies quantization errors non-linearly. For example, when x changes from 10.0 to 10.1 (1% change), $exp(x)$ changes from 22,026 to 24,343 (10.5% change). This sensitivity propagates through the attention computation, causing significant accuracy degradation under aggressive quantization.

Furthermore, Transformer's residual connections accumulate quantization errors across layers. For an L-layer Transformer with per-layer quantization error ε_l, the total accumulated error is approximately $\approx \sum_{l=1}^{L} \varepsilon_l$. With four layers and approximately 0.4 dB error per layer under INT8, the total approaches 1.6 dB—consistent with our experimental observation of 1.7 dB loss.

In contrast, CNN operations involve primarily linear transformations (convolutions) and piecewise-linear activations (ReLU). These operations are inherently more robust to quantization since small input perturbations produce proportionally small output changes without exponential amplification.

6.2. Practical considerations for deployment

Several practical factors influenced our deployment strategy. First, the choice of BF16 for Transformer layers rather than FP16 was motivated by dynamic range considerations. FP16's limited 5-bit exponent restricts dynamic range to $\pm 6.5 \times 10^4$, which can cause overflow in attention score computation. BF16's 8-bit exponent matches FP32's range[33] ($\pm 3.4 \times 10^{38}$), preventing numerical issues while halving memory requirements.

Second, the block-wise computation strategy required careful implementation to avoid performance degradation. Memory allocation and deallocation must be synchronized with computation to prevent fragmentation. We implemented a memory pool that pre-allocates buffers for all blocks and manages them through a lightweight scheduler, achieving consistent latency across inference iterations.

Third, protection against extreme heat and dust ingress proved critical in desert deployment. The edge computing module generates heat during intensive NPU operations while facing high ambient temperatures. Our hardware design includes enhanced heat dissipation optimized for high-temperature environments and appropriate sealing/filtration against fine sand, maintaining stable operating temperatures between −20 °C and +85 °C at the die level.

6.3. Comparison with cloud-based processing

While cloud-based processing can leverage more powerful hardware, edge deployment offers critical advantages for desert monitoring: (i) Latency—edge processing achieves 68 ms vs. seconds for round-trip cloud communication; (ii) Connectivity—desert stations often have intermittent or low-bandwidth communication links; (iii) Power—transmitting raw data consumes significantly more power than local processing; and (iv) Autonomy—edge systems continue operating during communication outages.

Our hybrid approach combines edge preprocessing with selective cloud upload. High-confidence detections (threshold >0.95) use edge results directly, while uncertain cases (0.5–0.95) are queued for cloud verification when connectivity permits. This strategy reduces data transmission by approximately 85% while maintaining detection reliability.

6.4. Generalization to other extreme environments

Although trained on Gansu desert data, the TFDenoiser-Edge framework is designed for generalization to other extreme environments. Time-frequency domain processing captures noise characteristics independent of specific geographic contexts. Preliminary tests on seismic data from polar environments and volcanic regions (characterized by tremor and explosion signals) showed promising results with minimal fine-tuning.[39,40]

For deployment in new environments, we recommend a transfer learning approach: (i) collect 2–4 weeks of continuous noise recordings from the target site; (ii) fine-tune only the decoder layers while freezing encoder and Transformer weights; and (iii) validate on held-out local data before deployment. This approach reduces adaptation time from weeks of full training to approximately 2–3 hours on the edge device itself.

7. Conclusion

This paper presents TFDenoiser-Edge, a hybrid CNN–Transformer framework[24,29] for real-time seismic denoising on edge devices deployed in extreme environments. The key contributions include a novel network architecture combining local and global feature modeling in the time-frequency domain, a mixed-precision quantization strategy differentially optimizing CNN and Transformer components, and a block-wise computation approach for memory-efficient edge deployment.

Extensive experiments demonstrated that TFDenoiser-Edge achieved 8.5 dB average SNR improvement while increasing P-wave and S-wave detection rates by 26 and 33 percentage points, respectively. The model runs in real time (68 ms latency) on edge NPUs, consuming less than 5 W of power. Successful deployment during the Gansu monitoring campaign validates the system's practical applicability for autonomous seismic monitoring in arid and desert regions.

This work represents the first successful deployment of Transformer-based deep learning models on seismic monitoring edge NPUs, providing both theoretical insights and practical solutions for intelligent geophysical sensing in resource-constrained environments. The proposed techniques are generalizable to other edge AI applications requiring efficient deployment of hybrid architectures.

Acknowledgments

None.

Funding

This work was supported by the National Key R&D Program of China (grant no. 2021YFA0716800), and the CAS Project for Young Scientists in Basic Research (grant no. YSBR-020).

Conflict of interest

The authors declare they have no competing interests.

Author contributions

Conceptualization: Qingfeng Xue
Formal analysis: Zesheng Yang, Qingfeng Xue, Tao Wang, Xiaoning Wang
Funding acquisition: Qingfeng Xue
Investigation: Zesheng Yang, Qingfeng Xue, Tao Wang
Methodology: Zesheng Yang, Qingfeng Xue
Validation: Qingfeng Xue
Visualization: Zesheng Yang, Qingfeng Xue, Xiaoning Wang
Writing–original draft: Zesheng Yang
Writing–review & editing: Qingfeng Xue

Availability of data

Data collected through the research presented in the paper are available upon request to the corresponding author.

References

1. Liu G, Chen X, Li J, Du J, Song J. Seismic noise attenuation using nonstationary polynomial fitting. *Appl Geophys*. 2011;8(1):18-26.

   doi: 10.1007/s11770-010-0244-2

2. Wu S, Wang Y, Di Z, Chang X. Random noise attenuation by 3D Multi-directional vector median filter. *J Appl Geophys*. 2018;159:277-284.

   doi: 10.1016/j.jappgeo.2018.09.021

3. Mousavi SM, Langston CA. Hybrid Seismic Denoising Using Higher-Order Statistics and Improved Wavelet Block Thresholding. *B Seismol Soc Am*. 2016;106(4):1380-1393.

   doi: 10.1785/0120150345

4. Yu Z, Abma R, Etgen J, Sullivan C. Attenuation of noise and simultaneous source interference using wavelet denoising. *Geophysics*. 2017;82(3):V179-V190.

   doi: 10.1190/geo2016-0240.1

5. Shao J, Wang Y, Liang X, Xue Qi, Liang E, Shi S. *Ji yu luan*

sheng wang luo de ren gong zhen yuan fen bu shi guang xian chuan gan shu ju zao sheng ya zhi [Siamese network based noise elimination of artificial seismic data recorded by distributed fiber-optic acoustic sensing]. *Chin J Geophys*. 2022;65(9):3599-3609. [In Chinese].

doi: 10.6038/cjg2022P0919

6. Li Y, Yu W, Zhang C, Yang B. Low-frequency noise suppression for desert seismic data based on a wide inference network. *J Geophys Eng*. 2019;16(4):801-810.

doi: 10.1093/jge/gxz051

7. Zhang S, Li Y. Seismic exploration desert noise suppression based on complete ensemble empirical mode decomposition with adaptive noise. *J Appl Geophys*. 2020;180:104055.

doi: 10.1016/j.jappgeo.2020.104055

8. Zhong T, Ye Y. MFIEN: Multi-scale feature interactive enhancement network for seismic data denoising in desert areas. *Sci Rep*. 2025;15:3979.

doi: 10.1038/s41598-025-87481-y

9. LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*. 2015;521(7553):436-444.

doi: 10.1038/nature14539

10. Mousavi SM, Beroza GC. Deep-learning seismology. *Science*. 2022;377(6607):eabm4470.

doi: 10.1126/science.abm4470

11. Zhu W, Beroza GC. PhaseNet: A Deep-Neural-Network-Based Seismic Arrival Time Picking Method. *Geophys J Int*. 2019;216(1):261-273.

doi: 10.1093/gji/ggy423

12. Mousavi SM, Ellsworth WL, Zhu W, Chuang LY, Beroza GC. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nat Commun*. 2020;11:3952.

doi: 10.1038/s41467-020-17591-w

13. Zheng Y, Wang Y, Liang X, *et al*. A deep learning approach for signal identification in the fluid injection process during hydraulic fracturing using distributed acoustic sensing data. *Front Earth Sci*. 2022;10:999530.

doi: 10.3389/feart.2022.999530

14. Zheng Y, Wang Y. Ground-penetrating radar wavefield simulation via physics-informed neural network solver. *Geophysics*. 2023;88(2):KS47-KS57.

doi: 10.1190/geo2022-0293.1

15. Wu S, Wang Y, Liang X. Joint denoising and classification network: Application to microseismic event detection in hydraulic fracturing distributed acoustic sensing monitoring. *Geophysics*. 2023;88(4):L53-L63.

doi: 10.1190/geo2022-0296.1

16. Li S, Yang X, Cao A, *et al*. SeisT: A foundational deep learning model for earthquake monitoring tasks. *IEEE Trans Geosci Remote Sens*. 2024;62:1-15.

doi: 10.1109/TGRS.2024.3371503

17. Shao J, Wang Y, Yao Y, Wu S, Xue Q, Chang X. Simultaneous denoising of multicomponent microseismic data by joint sparse representation with dictionary learning. *Geophysics*. 2019;84(5):KS155-KS172.

doi: 10.1190/geo2018-0512.1

18. Zhu W, Mousavi SM, Beroza GC. Seismic Signal Denoising and Decomposition Using Deep Neural Networks. *IEEE Trans Geosci Remote Sens*. 2019;57(11):9476-9488.

doi: 10.1109/TGRS.2019.2926772

19. Quinones L, Tibi R. Denoising Seismic Waveforms Using a Wavelet-Transform-Based Machine-Learning Method. *B Seismol Soc Am*. 2024;114(4):1777-1788.

doi: 10.1785/0120230304

20. Dantas PV, Sabino Da Silva W, Cordeiro LC, Carvalho CB. A comprehensive review of model compression techniques in machine learning. *Appl Intell*. 2024;54(22):11804-11844.

doi: 10.1007/s10489-024-05747-w

21. Ngo D, Park HC, Kang B. Edge Intelligence: A Review of Deep Neural Network Inference in Resource-Limited Environments. *Electronics*. 2025;14(12):2495.

doi: 10.3390/electronics14122495

22. Ross ZE, Meier MA, Hauksson E, Heaton TH. Generalized Seismic Phase Detection with Deep Learning. *B Seismol Soc Am*. 2018;108(5A):2894-2901.

doi: 10.1785/0120180080

23. Perol T, Gharbi M, Denolle M. Convolutional neural network for earthquake detection and location. *Sci Adv*. 2018;4(2):e1700578.

doi: 10.1126/sciadv.1700578

24. Vaswani A, Shazeer N, Parmar N, *et al*. Attention Is All You Need. *arXiv*. Preprint posted online 2017.

doi: 10.48550/arXiv.1706.03762

25. Seemakhupt K, Liu S, Khan S. EdgeRAG: Online-Indexed RAG for Edge Devices. *arXiv*. Preprint posted online 2024.

doi: 10.48550/arXiv.2412.21023

26. Zhang F, Zhang C, Guan J, *et al*. Breaking the Edge: Enabling Efficient Neural Network Inference on Integrated Edge Devices. *IEEE Trans Cloud Comput*. 2025;13(2):694-710.

doi: 10.1109/TCC.2025.3559346

27. Jacob B, Kligys S, Chen B, *et al*. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In: *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition;* June 18-23

2018; Salt Lake City, UT, USA; 2018:2704-2713.

doi: 10.1109/cvpr.2018.00286

28. Howard AG, Zhu M, *et al*. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv*. Preprint posted online 2017.

doi: 10.48550/arXiv.1704.04861

29. Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF, eds. *Proceedings of the Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Vol 9351.* Lecture Notes in Computer Science. MICCAI 2015; October 5-9 2015; Munich, Germany: Springer International Publishing; 2015:234-241.

doi: 10.1007/978-3-319-24574-4_28

30. Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: Bach F, Blei D, eds. *Proceedings of the 32nd International Conference on Machine Learning.* Volume 37: International Conference on Machine Learning; July 7-9 2015; Lille, France; 2015:448-456. https://proceedings.mlr.press/v37/ioffe15.html

31. He K, Zhang X, Ren S, Sun J. Deep Residual Learning for Image Recognition. In: *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* June 27-30 2016; Las Vegas, NV, USA; 2016:770-778.

doi: 10.1109/cvpr.2016.90

32. Zhou P, Xie X, Lin Z, Yan S. Towards Understanding Convergence and Generalization of AdamW. *IEEE Trans Pattern Anal Mach Intell*. 2024;46(9):6486-6493.

doi: 10.1109/TPAMI.2024.3382294

33. Micikevicius P, Narang S, Alben J, *et al*. Mixed Precision Training. In: *Proceedings of the 2018 International Conference on Learning Representations.* 6th International Conference on Learning Representations; April 30-May 3 2018; Vancouver, Canada; 2018. https://openreview.net/forum?id=r1gs9JgRZ

34. Allen RV. Automatic earthquake recognition and timing from single traces. *B Seismol Soc Am*. 1978;68(5):1521-1532.

doi: 10.1785/BSSA0680051521

35. Saad OM, Chen Y. Deep denoising autoencoder for seismic random noise attenuation. *Geophysics*. 2020;85(4):V367-V376.

doi: 10.1190/geo2019-0468.1

36. Yang L, Liu X, Zhu W, Zhao L, Beroza GC. Toward improved urban earthquake monitoring through deep-learning-based noise suppression. *Sci Adv*. 2022;8(15):eabl3564.

doi: 10.1126/sciadv.abl3564

37. Yin J, Denolle MA, He B. A multitask encoder–decoder to separate earthquake and ambient noise signal in seismograms. *Geophys J Int*. 2022;231(3):1806-1822.

doi: 10.1093/gji/ggac290

38. Lomax A, Michelini A, Curtis A. Earthquake Location, Direct, Global-Search Methods. In: Meyers, R, ed. *Encyclopedia of Complexity and Systems Science*. New York, NY, USA: Springer New York; 2009:2449-2473.

doi: 10.1007/978-0-387-30440-3_150

39. Lapins S, Butcher A, Kendall JM, *et al*. DAS-N2N: machine learning distributed acoustic sensing (DAS) signal denoising without clean data. *Geophys J Int*. 2023;236(2):1026–1041.

doi: 10.1093/gji/ggad460

40. Fernández-Carabantes J, Titos M, D'Auria L, García J, García L, Benítez C. RNN-DAS: A New Deep Learning Approach for Detection and Real-Time Monitoring of Volcano-Tectonic Events Using Distributed Acoustic Sensing. *JGR Solid Earth*. 2025;130(9):e2025JB031756.

doi: 10.1029/2025JB031756