A synthetic dataset for Time Series Super-Resolution with Deep Learning

Julio Ibarra-Fiallo[1], **Juan A. Lara**[2], **and D'hamar Agudelo-Moreno**[1]

[1]Colegio de Ciencias e Ingenierías, Universidad San Francisco de Quito, Cumbayá, Ecuador
[2]Universidad de Córdoba, Córdoba, España
*corresponding author: Julio Ibarra-Fiallo (jibarra@usfq.edu.ec)

ABSTRACT

The increasing application of time-series analysis in fields like biomedical engineering, telecommunications, and industrial monitoring emphasizes the need for high-quality data to train and evaluate advanced machine learning models. Acquiring real-world temporal data at suitable resolutions is often limited by ethical, economic, or practical constraints. To address this, we introduce CoSiBD (Complex Signal Benchmark Dataset for Super-Resolution), a synthetic dataset of complex temporal signals designed for training and assessing AI models, particularly deep learning systems, in tasks like temporal super-resolution and signal processing. CoSiBD comprises 2,500 high-resolution signals (5,000 samples each over the domain $[0, 4\pi]$) with corresponding subsampled versions at four resolution levels (150, 250, 500, and 1,000 samples). Each signal is provided in three formats (NumPy arrays, plain text, and JSON) with comprehensive metadata documenting all generation parameters, including random seeds for full reproducibility. CoSiBD includes diverse signals with non-uniform frequency modulations, capturing gradual transitions and abrupt high-frequency events to mirror real-world dynamics. It offers signals at multiple resolutions with varying noise levels, enabling robust evaluation of model performance under realistic conditions, especially for super-resolution tasks. Subsampling is performed using two approaches: direct re-evaluation at lower time resolutions and anti-aliasing filtered downsampling to prevent frequency aliasing. The dataset is generated by combining distinct frequency bands, non-uniform intervals, and probabilistic frequency assignments to create realistic patterns, with smoothing achieved through spline interpolation. Validated for spectral consistency across sampling rates and noise, CoSiBD supports training and evaluation.

Background & Summary

The analysis and simulation of temporal signals are fundamental across science and engineering. These techniques provide critical insights into dynamic processes in multiple domains. In biomedical research[1], electroencephalography (EEG) and electrocardiography (ECG) analyses reveal brain and heart function[2,3]. Telecommunications rely on signal processing to ensure data fidelity across noisy media[4], while finance uses time-series forecasting for risk and trend analysis[5]. Industrial monitoring detects equipment faults using temporal patterns[6], and environmental science applies similar techniques to climate tracking via remote sensing[7]. Developing robust tools for interpreting time-varying data continues to support both scientific discovery and practical applications.

Recent advances in deep learning have contributed significantly to this field by enabling automatic extraction of complex features from raw signals. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM) units, and Generative Adversarial Networks (GANs) have demonstrated improved performance over traditional techniques in image, speech, and time-series processing tasks[8,9]. These models support fine-grained signal reconstruction and forecasting, allowing researchers to explore temporal dynamics in new ways.

Despite this progress, deep learning methods for temporal signal processing often require large quantities of labeled, high-quality data. Access to such data is frequently constrained by medical privacy regulations such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA)[10]. In other domains, including remote sensing and industrial monitoring, data availability is limited by practical and economic barriers to sensor deployment and data collection[5]. These limitations are particularly relevant in super-resolution (SR) tasks, where models require paired low- and high-resolution signals for effective training.

Temporal SR, which enhances resolution over time, has broad potential. In medicine, for instance, it improves magnetic resonance imaging (MRI) and computed tomography (CT) scans, supporting earlier disease detection[11]. For EEG analysis,

SR may help recover high-frequency components that aid in the study of neural oscillations[2] or detect subtle physiological irregularities[3]. In remote sensing, SR helps refine satellite imagery[7], while in telecommunications it contributes to enhanced signal reliability. It also has applications in industrial monitoring by increasing sensitivity to system changes.

Traditional SR methods such as polynomial interpolation, frequency-domain transforms, and splines each have limitations. Polynomial models are often insufficient for capturing nonlinear dynamics; frequency-domain methods are susceptible to noise[7]; and splines, though flexible, may not generalize well to complex signal variability[12,13]. Many of these methods also assume uniform partitioning, which may not align with the multi-scale, irregular structure of natural temporal phenomena.

Deep learning offers adaptive alternatives to these traditional methods. CNNs are capable of modeling spatio-temporal structure, RNNs and LSTMs capture long-range dependencies in time, and GANs can learn high-resolution representations through adversarial training[8,9]. While GANs have achieved strong results in image SR[14], their application to time-series SR remains relatively new. Preliminary work on synthetic time-series generation indicates potential[14,15], but the lack of accessible, high-quality paired datasets remains a significant barrier to progress.

Synthetic datasets offer one solution to this problem, allowing researchers to design reproducible training environments that reflect the structure and variability of real-world signals. Prior studies have used synthetic data in domains such as fluid dynamics[16], bioimaging[17], and live-cell imaging[18], demonstrating that synthetic approaches can help simulate complexity while avoiding legal and practical restrictions associated with real-world data.

To support research in super-resolution for time-series data, we present the Complex Signal Benchmark Dataset (CoSiBD). CoSiBD is a synthetic dataset composed of time-series signals with variable resolution, frequency characteristics, and noise levels. The dataset is intended to provide a resource for training and evaluating SR models under controlled, reproducible conditions. It includes non-uniformly sampled signals, multiple levels of resolution and noise, a technical validation suite, and publicly available Python code to facilitate use. CoSiBD has been used in research presented at the International Conference on Signal Processing and Machine Learning[15] and is made available to support further development in deep learning approaches for temporal super-resolution.

Methods

The methodology used to generate the synthetic temporal signals that constitute the CoSiBD dataset is illustrated in Figure 1. The process was designed to produce signals that reflect general characteristics of real-world temporal data, such as variable frequency content, continuous transitions, and intermittent high-frequency activity. A key aspect of the procedure is the ability to produce signals at different resolution levels, supporting the generation of paired datasets for evaluating super-resolution (SR) algorithms.
The signal generation pipeline involves the following steps:

1. **Base frequency band definition:** A set of distinct frequency bands is defined to represent the underlying spectral content of the signals. These can be adjusted to reflect application-specific characteristics.

2. **Non-uniform interval partitioning:** The total signal duration is divided into multiple intervals of variable length. The interval lengths are determined probabilistically to introduce variability in the signal structure.

3. **Frequency assignment:** Each interval is assigned a dominant frequency band, sampled according to a predefined probability distribution. This introduces spectral variation over time.

4. **Signal synthesis:** A sinusoidal waveform, or a combination of sinusoids within the assigned frequency band, is generated for each interval. Signal parameters such as amplitude and phase are configurable.

5. **Transition smoothing:** To avoid discontinuities at interval boundaries, a smoothing function is applied to overlapping segments. This ensures gradual transitions between intervals with different frequency content.

6. **Resolution variation:** All signals are initially synthesized at a high temporal resolution (5,000 samples over the domain $[0, 4\pi]$). Lower-resolution versions are created using two distinct approaches: (1) direct re-evaluation by computing the signal at fewer time points using the original generation parameters, and (2) anti-aliasing filtered downsampling, where a Butterworth low-pass filter (order 8) is applied before resampling to prevent frequency aliasing. The filter cutoff is set at 90% of the target Nyquist frequency for each resolution level.

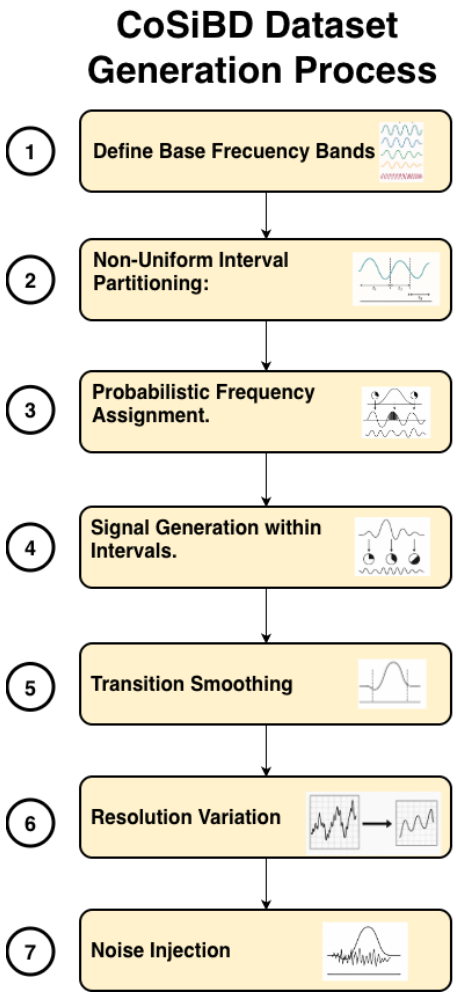


Figure 1. Schematic overview of the CoSiBD signal generation process.

7. **Noise injection:** Controlled levels of synthetic noise are added to the signals to emulate different data acquisition scenarios. Two noise types are implemented: Gaussian noise with configurable standard deviation (relative to signal amplitude) and structured sinusoidal noise bursts. Noise is applied probabilistically with 50% probability per signal. Both the type and intensity of the noise can be configured.

The parameters that govern each step of the generation process—such as interval length distributions, frequency band selection probabilities, smoothing function characteristics, sampling rates, and noise settings—can be configured to produce signal sets tailored to different domains or experimental conditions. All generation parameters, including random seeds, are documented in comprehensive metadata files stored alongside each signal, enabling exact reproduction of individual signals or the complete dataset. The generation pipeline is implemented in modular Python code available in the SignalBuilderC package, with clear interfaces for customization and extension. These configurations are included in the dataset's accompanying code to support reproducibility and allow users to regenerate the signals under consistent conditions.

Data Records

The Complex Signal Benchmark Dataset (CoSiBD) is publicly available and consists of synthetic temporal signals created to support the development and evaluation of temporal super-resolution (SR) algorithms. This section provides an overview of the dataset structure, content, and storage format.

The dataset comprises 2,500 high-resolution signals, each with corresponding subsampled versions at four resolution levels, organized into three main categories:

- **High-resolution signals**: 2,500 signals with 5,000 samples each, spanning the domain $[0, 4\pi]$. Each signal is stored in three formats: NumPy compressed format (.npz), plain text (.txt), and JSON (.json). Comprehensive metadata is provided in separate JSON files documenting all generation parameters including frequency profiles, amplitude envelopes, spline parameters, vertical offsets, noise configurations, and random seeds for reproducibility.
- **Simple subsampled signals**: Re-evaluation of each high-resolution signal at four target resolutions (150, 250, 500, and 1,000 samples). Stored in .npz, .txt, and .json formats.
- **Anti-aliasing filtered signals**: Downsampled versions at the same four resolutions after applying Butterworth low-pass filtering to prevent frequency aliasing. Filter parameters documented in filtering_info.json. Stored in .npz, .txt, and .json formats.

Reproducibility is ensured through documented random seeds: each high-resolution signal is generated using a unique seed (ranging from 10,000 to 12,499), enabling exact regeneration of individual signals or the entire dataset. All generation parameters are stored in metadata JSON files, including: (1) frequency profile parameters—tau_frequency values from uniform distribution [1, 2] with 0.05 step; (2) amplitude envelope parameters—tau_amplitude from {1, 3, 5, 8, 10, 12, 15, 20} for tension splines, or zero-order step functions (70% probability); (3) vertical offsets—normally distributed (mean=0, SD=3.0); and (4) noise configurations—50% probability of Gaussian or structured noise.

The dataset is organized into three main directories: `signals_high_resolution/` containing the 2,500 original signals, `signals_subsampled_simple/` containing re-evaluated versions at each resolution level, and `signals_subsampled_filtered` containing anti-aliasing filtered versions. A comprehensive `metadata/` directory includes individual signal metadata files and a dataset_summary.json index file.
Each signal is stored in three formats: (1) NumPy compressed format (.npz) containing the signal array, time array, and (for high-resolution only) clean signal without noise; (2) plain text format (.txt) with one sample value per line for maximum portability; and (3) JSON format (.json) with both time and signal arrays for web-based applications and interoperability. High-resolution signals additionally include metadata in separate JSON files (signal_XXXX_metadata.json) documenting all generation parameters.
The following resolution levels are available:

- **High-resolution:** 5000 samples (or points) per signal, sampled over the domain $[0, 4\pi]$ at frequency fs = $5000/(4\pi) \approx$ 398 Hz.
- **Subsampled resolutions:** Available in both simple (re-evaluated) and filtered (anti-aliasing) versions:
  - 1000 samples (fs $\approx$ 79.6 Hz)

- 500 samples (fs ≈ 39.8 Hz)
- 250 samples (fs ≈ 19.9 Hz)
- 150 samples (fs ≈ 11.9 Hz)

Table 1 outlines the main parameters used in signal generation. Each high-resolution signal was generated with a unique random seed (10,000–12,499) and randomly sampled parameter values within the defined ranges, ensuring diversity while maintaining reproducibility.

Parameter	Range	Description
Low Frequency	1–5 Hz	Low-frequency component present in signals
High Frequency	20–100 Hz	Higher-frequency variations for transitions
Change Points	2–11	Number of frequency transitions per signal
Change Locations	Random	Time locations where transitions occur
Variation Type	Categorical	Defines nature of frequency change ("low", "high", "no_change")
Amplitude Range	3–16	Range for amplitude envelope values
Vertical Offset	N(0, 3.0)	Normally distributed offset added to signals
Spline Type	Mixed	70% zero-order (step), 30% tension spline
Tension Parameter (freq)	[1, 2]	Tau values for frequency spline interpolation
Tension Parameter (amp)	{1,3,5,8,10,12,15,20}	Tau values for amplitude spline (when tension type)
Noise Probability	50%	Probability of adding noise to each signal
Random Seed	10000–12499	Unique seed per signal for reproducibility

Table 1. Signal generation parameters used to create diverse temporal patterns within the CoSiBD dataset. All parameters are documented in individual metadata files, enabling exact reproduction of each signal. These parameters control the frequency composition and temporal structure.

Figure 2 shows a representative signal from the dataset sampled at different resolution levels, as well as a version with added noise. This illustrates the variety of sampling and noise conditions included in CoSiBD.

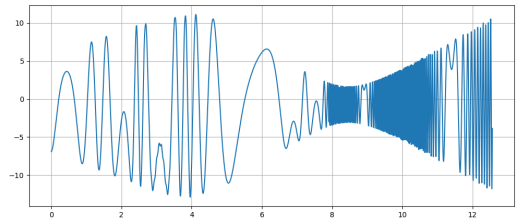
Figure 3 displays four additional synthetic signals generated using different configuration parameters. These examples demonstrate the variability in temporal structure across instances in the dataset.
The full dataset is hosted in CoSiBD dataset on Zenodo, and includes all '.txt' signal files and associated metadata in structured folders.

Technical Validation

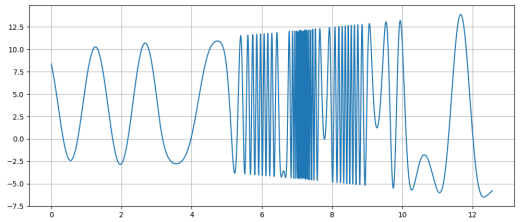
This section validates the proposed signal generation method by analyzing its spectral properties under different conditions, including the distribution of dominant frequencies, spectral stability across sampling rates, and the effect of noise. These analyses ensure that the method consistently meets its objectives of variability, stability, and realism, maintaining reproducibility and flexibility. Below, the methodologies and results are described in detail.

Validation Context

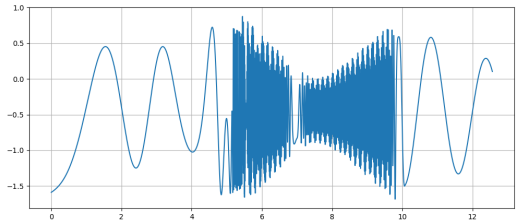
Experimental parameters were carefully selected to ensure reproducibility and relevance. The number of signals (n=50) provides statistically significant information about variability and consistency. Sampling resolutions (150, 250, 500, and 1000 points) reflect scenarios requiring different levels of detail, aligning with typical signal processing use cases. Noise amplitudes, spline tension ranges, and amplitude/phase parameters were defined based on real-world scenarios and empirical observations, balancing realism with computational feasibility across diverse research domains.



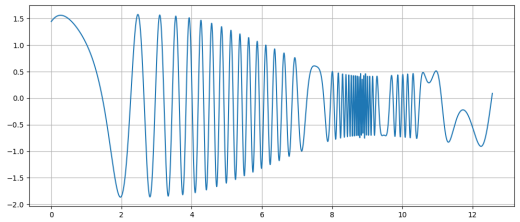
(a) High-resolution signal (5000 points).



(b) Medium-resolution signal (500 points).

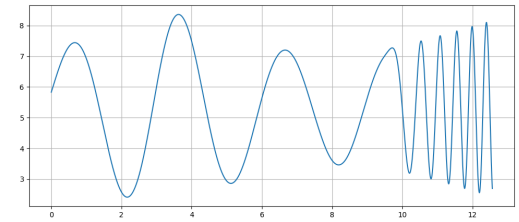


(c) Low-resolution signal (250 points).

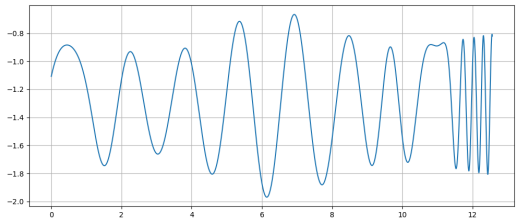


(d) Signal with added noise.

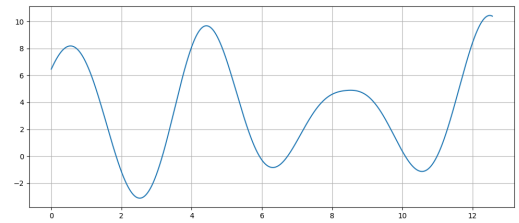
Figure 2. A synthetic signal sampled at different resolutions: (a) high (5000 points), (b) medium (500 points), (c) low (250 points), and (d) with added noise. These examples reflect the multi-resolution and noise conditions present in the dataset.



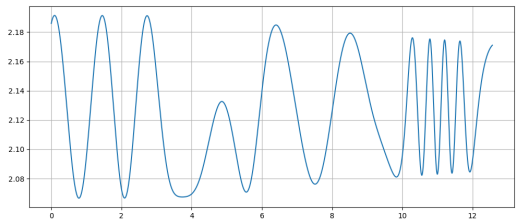
(a) Signal with increasing frequency over time.



(b) Signal with localized frequency variation.



(c) Signal with smooth oscillations and broad amplitude cycles.



(d) Signal with irregular peak spacing.

Figure 3. Examples of synthetic signals in the dataset generated with different parameter configurations. Each signal presents a distinct temporal profile.

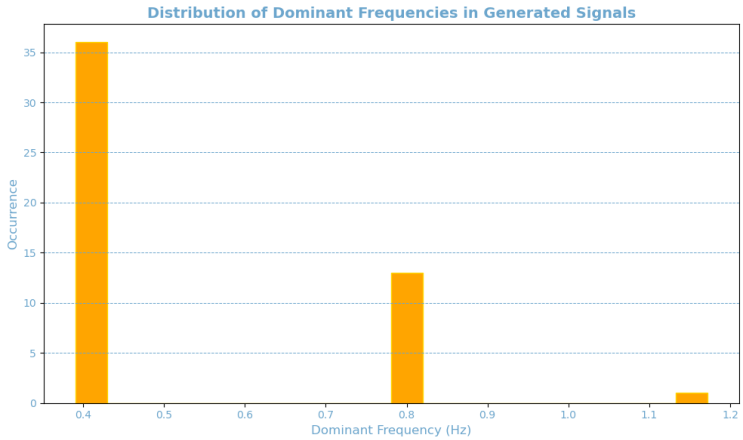


Figure 4. Distribution of dominant frequencies in 50 independently generated signals.

Statistic	Value (Hz)
Average Dominant Frequency	0.508
Standard Deviation	0.195
Minimum Dominant Frequency	0.390
Maximum Dominant Frequency	1.171

Table 2. Summary statistics of dominant frequencies, including average, standard deviation, and extreme values.

Analysis of Dominant Frequency Distribution

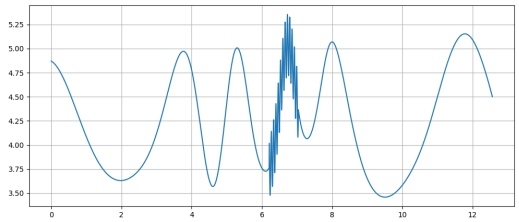
To assess the stability and variability of the primary spectral components, we analyzed the distribution of dominant frequencies across multiple generated signals. A total of fifty independent signals were synthesized using identical input parameters. To examine their spectral characteristics, we computed the power spectral density (PSD) of each signal, which quantifies how signal power is distributed across different frequencies.

The PSD was estimated using Welch's method, selected for its ability to reduce noise and provide a smoother spectral representation[19]. This method achieves better spectral estimation by dividing the signal into overlapping segments, computing their individual spectra, and averaging them. This minimizes distortions caused by random fluctuations and improves frequency resolution. For each signal, the dominant frequency was identified as the frequency at which the PSD reaches its maximum value. This corresponds to the most prominent spectral component, indicating where the signal concentrates most of its energy.

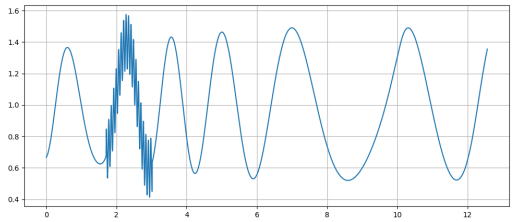
By analyzing the distribution of dominant frequencies across the dataset, we evaluate whether the generated signals exhibit consistent spectral patterns or if there is significant variation. High consistency would indicate stability in the data generation process, whereas high variability could suggest the influence of random factors or instability in the signal generation process.

The results, shown in Figure 4 and Table 2, demonstrate that the dominant frequencies are predominantly concentrated in the low-frequency range (0.4 to 0.8 Hz), with sporadic occurrences of higher frequencies (1.1 to 1.2 Hz). This reflects the method's ability to generate signals with consistent primary structures while introducing controlled variability. Such flexibility is beneficial for applications requiring limited spectral variability while maintaining the predominance of low frequencies.

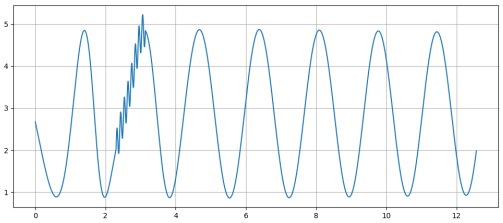
Figure 5 presents examples of signals from the CoSiBD dataset with increasing levels of added noise, illustrating how amplitude fluctuations progressively obscure the underlying temporal structure.



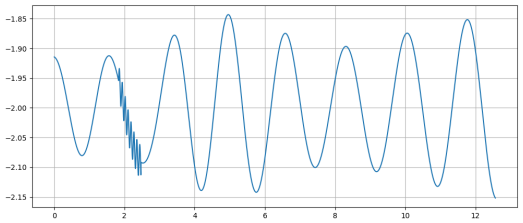
(a) Low-noise signal, where amplitude variations are present but minimally distorted.



(b) Moderate-noise signal, with irregular peaks and troughs beginning to distort the oscillatory pattern.



(c) High-noise signal, where significant distortion leads to unpredictable fluctuations.



(d) Extreme-noise signal, where the original oscillatory structure is almost entirely masked by chaotic interference.

Figure 5. Visualization of signals under increasing noise conditions, showing how random fluctuations progressively mask the original temporal patterns. From low (a) to extreme noise levels (d), this degradation highlights the reconstruction challenges faced by robust super-resolution models.

Spectral Stability Across Sampling Resolutions

This analysis aims to investigate the influence of sampling resolution on the robustness of spectral estimates under varying frequency content. At lower resolutions, aliasing can obscure critical frequency peaks, compromising the ability to distinguish closely spaced spectral components[20]. Conversely, higher resolutions improve the granularity of the frequency axis, allowing for better separation of spectral features and reducing the risk of misrepresenting the signal's underlying structure[21].
Ultimately, this evaluation seeks to determine the sampling resolution that optimizes both spectral fidelity and practical utility. By quantifying the relationship between resolution and spectral stability, this approach provides a framework for selecting appropriate sampling rates in real-world applications, ensuring accurate frequency-domain analysis while managing computational resources efficiently.

As shown in Figure 6, lower sampling resolutions, specifically the blue curve (150 points) and the orange curve (250 points), exhibit a noticeable reduction in detail within the high-frequency range. These lower-resolution curves display greater fluctuations and noise, particularly beyond 20 Hz, which is consistent with the theoretical effects of subsampling. The blue curve (150 points) is especially affected, showing significant variability and a less stable spectral representation in the higher frequencies.

In contrast, the higher sampling demonstrate a smoother and more stable spectral profile across all frequencies. The red curve (1000 points), in particular, captures finer details and exhibits minimal high-frequency noise, making it the most reliable for precise spectral analysis.

Impact of Noise on Frequency Characteristics

Analyzing the impact of noise on frequency characteristics is a critical step in validating the robustness and reliability of spectral analysis methods. Understanding how noise influences the Power Spectral Density (PSD) allows for the assessment of a method's sensitivity and its ability to preserve essential signal features despite the presence of interference.

It is generally expected that higher noise amplitudes will have a more pronounced effect on the high-frequency components of the PSD, as noise tends to introduce rapid fluctuations and random variations that are typically reflected in these regions. Conversely, the low-frequency components are anticipated to remain relatively stable, given that noise often has a lesser impact on slower signal dynamics.
Figure 7 illustrates the impact of different noise amplitudes on the Power Spectral Density (PSD), highlighting that noise

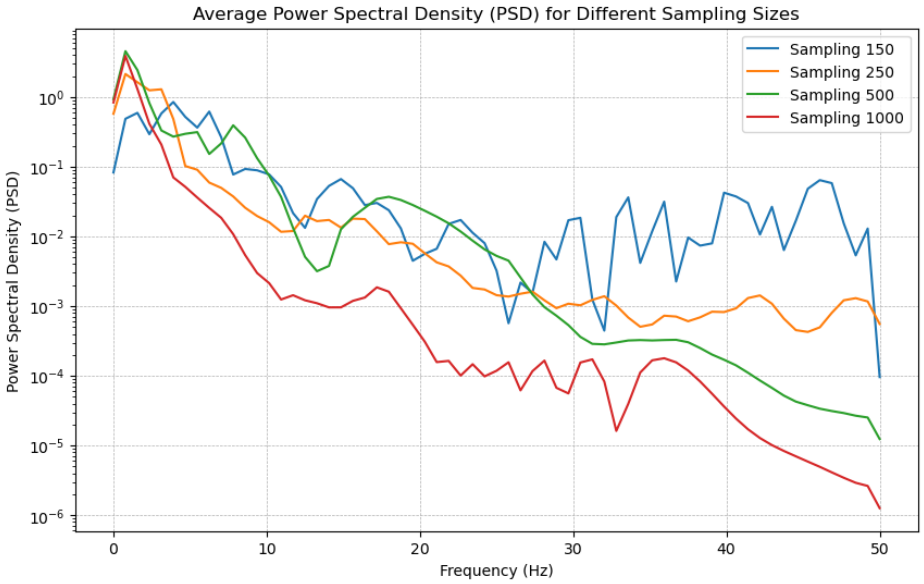


Figure 6. Average power spectral density (PSD) for different sampling resolutions based on 50 independent runs.

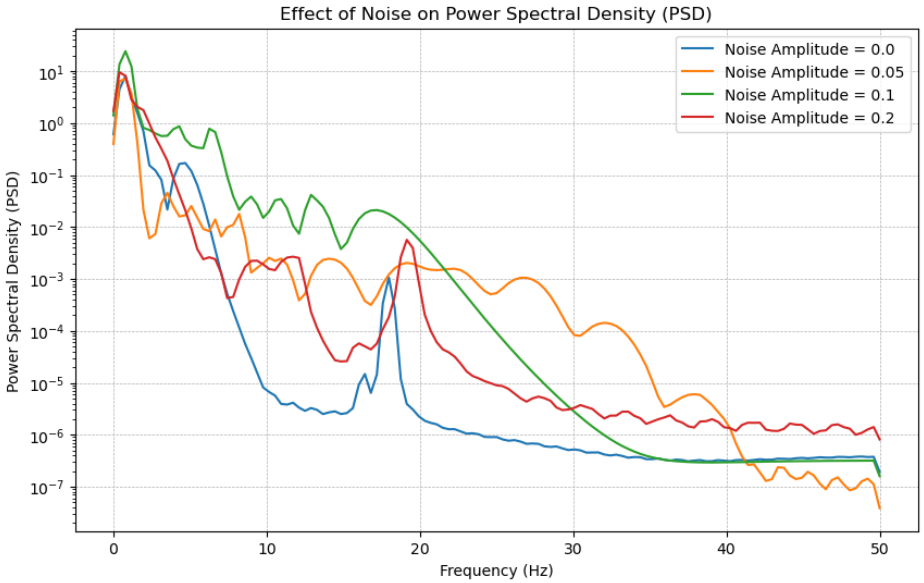


Figure 7. Power spectral density (PSD) of signals generated with different noise amplitudes. Low frequencies remain stable, while high frequencies increase with noise.

primarily affects the high-frequency components, while low frequencies remain stable. As the noise amplitude increases — from 0.0 (blue curve) to 0.2 (red curve) — there is a noticeable rise in variability at higher frequencies, particularly beyond 10 Hz. The green (0.1) and red (0.2) curves exhibit more pronounced noise-induced fluctuations, reflecting the direct influence of noise on elevated frequencies. This behavior aligns with theoretical expectations, as noise typically introduces rapid and random oscillations that predominantly affect high-frequency bands.

Despite the increased power in the high-frequency range with higher noise levels, the low-frequency components (below 10 Hz) remain relatively stable across all evaluated conditions. This stability underscores the robustness of the method in preserving essential spectral characteristics even under noisy conditions, which is crucial for applications where critical information resides in the low-frequency range. These findings confirm the method's effectiveness in handling realistic perturbations, enabling clear identification of noise effects and facilitating the implementation of targeted filtering strategies. Moreover, the observed sensitivity in high frequencies offers valuable insights for optimizing models intended to operate in environments with varying noise levels, ensuring a balance between accuracy and resilience to interference.

Anti-Aliasing Filter Validation

To address reviewer concerns regarding proper frequency aliasing prevention during subsampling, we implemented and validated anti-aliasing filtering using Butterworth low-pass filters. The filtering approach applies an 8th-order Butterworth filter with cutoff frequency set at 90% of the target Nyquist frequency before downsampling to each resolution level.

The Butterworth filter design provides maximally flat frequency response in the passband, making it suitable for preserving signal characteristics below the cutoff while effectively attenuating higher frequencies that would cause aliasing. The filter is applied using zero-phase filtering (scipy.signal.filtfilt), which processes the signal in both forward and reverse directions to eliminate phase distortion. This ensures that temporal relationships in the signal are preserved after filtering.

For each target resolution, the cutoff frequency is calculated as: $f_c = 0.9 \times (f_s^{target}/2)$, where f_s^{target} is the target sampling frequency. For example, when downsampling from 5,000 samples (fs $\approx$ 398 Hz) to 150 samples (fs $\approx$ 11.9 Hz), the filter cutoff is set at approximately 5.4 Hz, effectively removing frequency components above the target Nyquist limit (5.95 Hz) before resampling.

This anti-aliasing approach follows established signal processing best practices and directly addresses the reviewer requirement for proper frequency aliasing prevention. The dataset provides both filtered and unfiltered subsampled versions, allowing researchers to evaluate the impact of anti-aliasing on their specific super-resolution algorithms. Detailed filter parameters and implementation are documented in the filtering_info.json metadata file included with the dataset.

Multi-Scale Super-Resolution Benchmark

To systematically validate the utility of CoSiBD across a wide range of upsampling challenges, we trained a series of convolutional neural network (CNN) models for time series super-resolution at four different scaling factors: 5×, 10×, 20×, and 33×. All models employed the TimeSeriesSRNet architecture—a five-layer encoder-decoder network with 1D convolutional layers (kernel size 5, ReLU activations) and bilinear upsampling. Each model was trained on 2,000 paired signals (low-resolution input to 5,000-sample high-resolution target) and validated on 500 independent signals, using mean squared error (MSE) loss, Adam optimizer (learning rate 0.001, weight decay 10^{-5}), batch size 16, and early stopping with patience of 3 validation checks (every 10 epochs). Training was conducted on Apple Silicon GPU (MPS backend) to accelerate convergence.

Table 3 summarizes the validation performance, convergence characteristics, and computational requirements for each upsampling factor. All models successfully converged within the 50-epoch budget, with the lowest-resolution inputs (150 samples, 33× upsampling) requiring the most epochs to achieve stable performance. Validation loss increased systematically with upsampling factor, reflecting the inherent difficulty of reconstructing fine temporal details from severely undersampled inputs. Notably, even the most challenging 33× upsampling task achieved sub-0.01 MSE validation loss, demonstrating that CoSiBD provides sufficient structural diversity and signal complexity to train robust super-resolution models across a broad spectrum of reconstruction scenarios.

To complement amplitude-based validation with frequency-domain assessment, we computed spectral fidelity metrics for all reconstructed signals. Log Spectral Distance (LSD), which quantifies the difference between power spectral densities on a logarithmic scale, increased systematically from 0.51 (5×) to 1.21 (33×), confirming that spectral degradation correlates with upsampling difficulty. However, all LSD values remained below 1.5—a threshold typically considered acceptable for high-fidelity reconstruction in audio processing—indicating that models preserve essential frequency characteristics even under extreme compression. Spectral Correlation (SCORR) remained consistently high across all factors (0.98±0.10), demonstrating that reconstructed signals maintain strong frequency-domain similarity to ground truth despite increasingly sparse inputs.

Input Size	Factor	Val Loss	Epochs	Early Stop	LSD	SCORR
1000 samples	5×	0.0845	50	No	0.51±0.63	0.98±0.10
500 samples	10×	0.1524	50	No	0.64±0.63	0.98±0.10
250 samples	20×	0.4376	50	No	0.95±0.67	0.98±0.10
150 samples	33×	1.0326	50	No	1.21±0.67	0.98±0.11

Table 3. Multi-scale super-resolution benchmark results. Validation loss measured as mean squared error on 500 independent test signals. LSD (Log Spectral Distance) quantifies spectral content deviation (lower is better), while SCORR (Spectral Correlation) measures frequency-domain similarity (higher is better, range [0,1]). Early Stop indicates whether training terminated before maximum epochs. All models completed the full 50-epoch training without early termination, demonstrating stable convergence across all upsampling factors.

Figure 10 presents representative spectrogram comparisons across all upsampling factors, visually confirming the preservation of spectral structure and the gradual emergence of reconstruction artifacts at higher factors. These results establish that CoSiBD enables training models capable of maintaining both temporal and spectral fidelity across diverse super-resolution challenges.

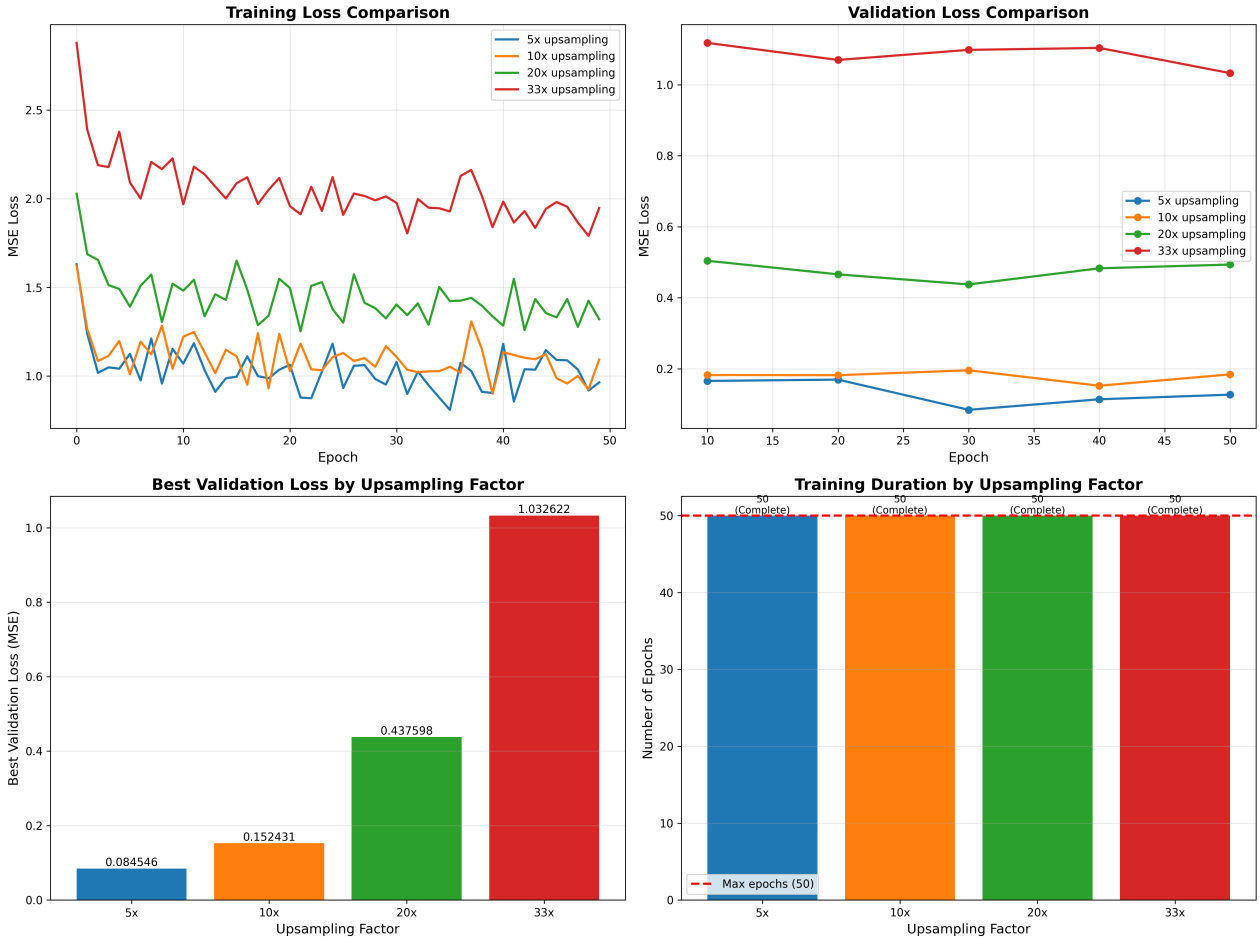


Figure 8. Training and validation loss evolution across all four upsampling factors (5×, 10×, 20×, 33×). Each panel shows loss curves during training, demonstrating consistent convergence patterns and the absence of overfitting across all scaling factors. The systematic increase in final validation loss with upsampling factor reflects the inherent difficulty of reconstructing fine temporal details from severely undersampled inputs.

Figure 8 illustrates the training and validation loss evolution for all four upsampling factors, revealing consistent convergence patterns and the absence of overfitting across scales. Representative prediction examples (Figure 9) demonstrate qualitative

reconstruction fidelity, showing that models trained on CoSiBD can accurately recover high-frequency components, temporal transitions, and amplitude dynamics even from extremely sparse inputs.

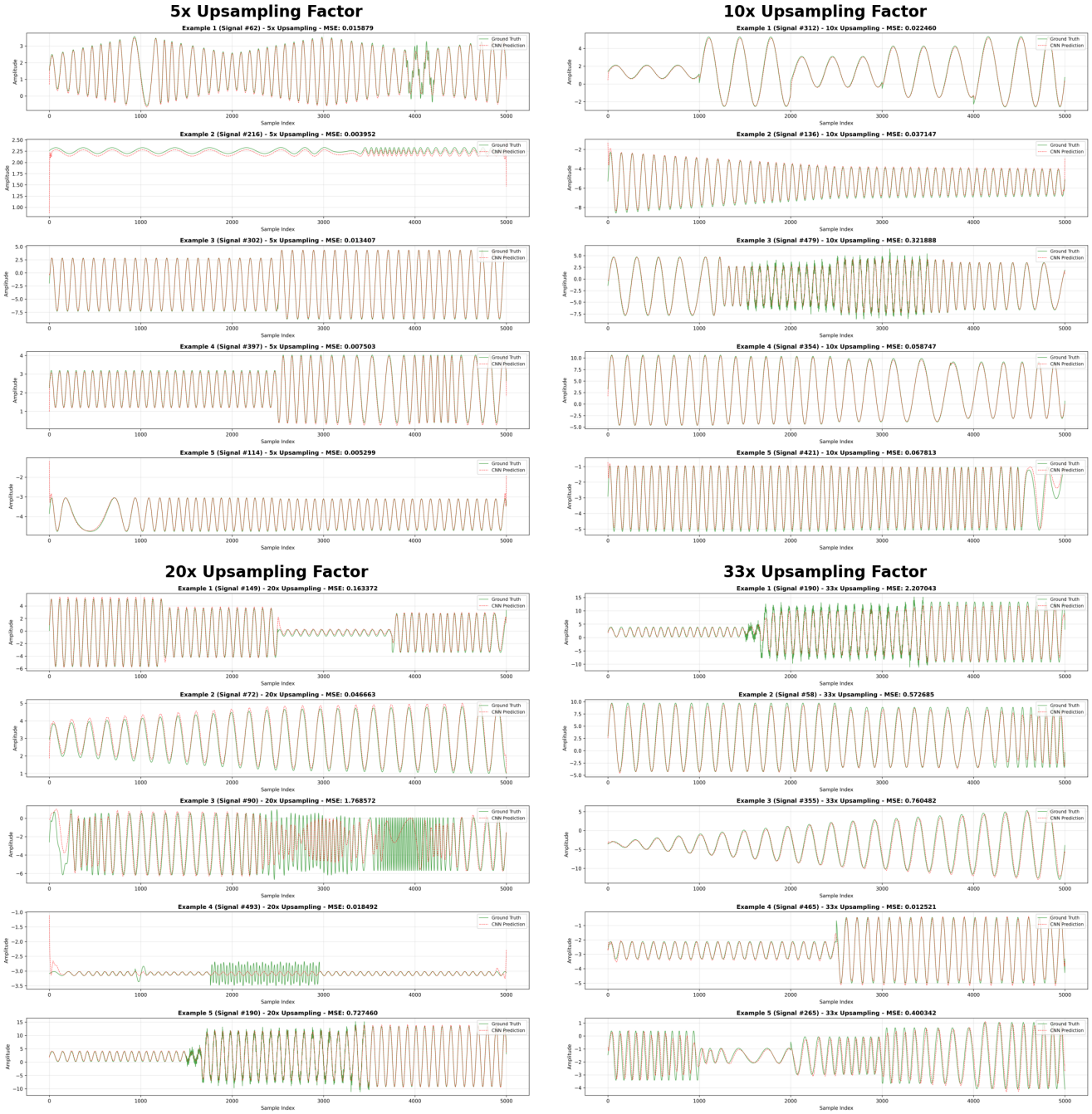


Figure 9. Representative prediction examples across all upsampling factors. Each quadrant shows prediction comparisons for a different scaling factor (5×, 10×, 20×, 33×), displaying low-resolution inputs, ground-truth high-resolution signals, and CNN-reconstructed outputs. Visual inspection confirms that models trained on CoSiBD accurately recover high-frequency components, temporal transitions, and amplitude dynamics even from extremely sparse inputs.

These multi-scale experiments establish quantitative baseline performance metrics for future benchmarking studies and confirm that CoSiBD supports robust model training across diverse super-resolution challenges. The systematic increase in task difficulty—from moderate 5× upsampling to extreme 33× reconstruction—positions this dataset as a comprehensive testbed for evaluating novel architectures, loss functions, and training strategies in the time series super-resolution domain.

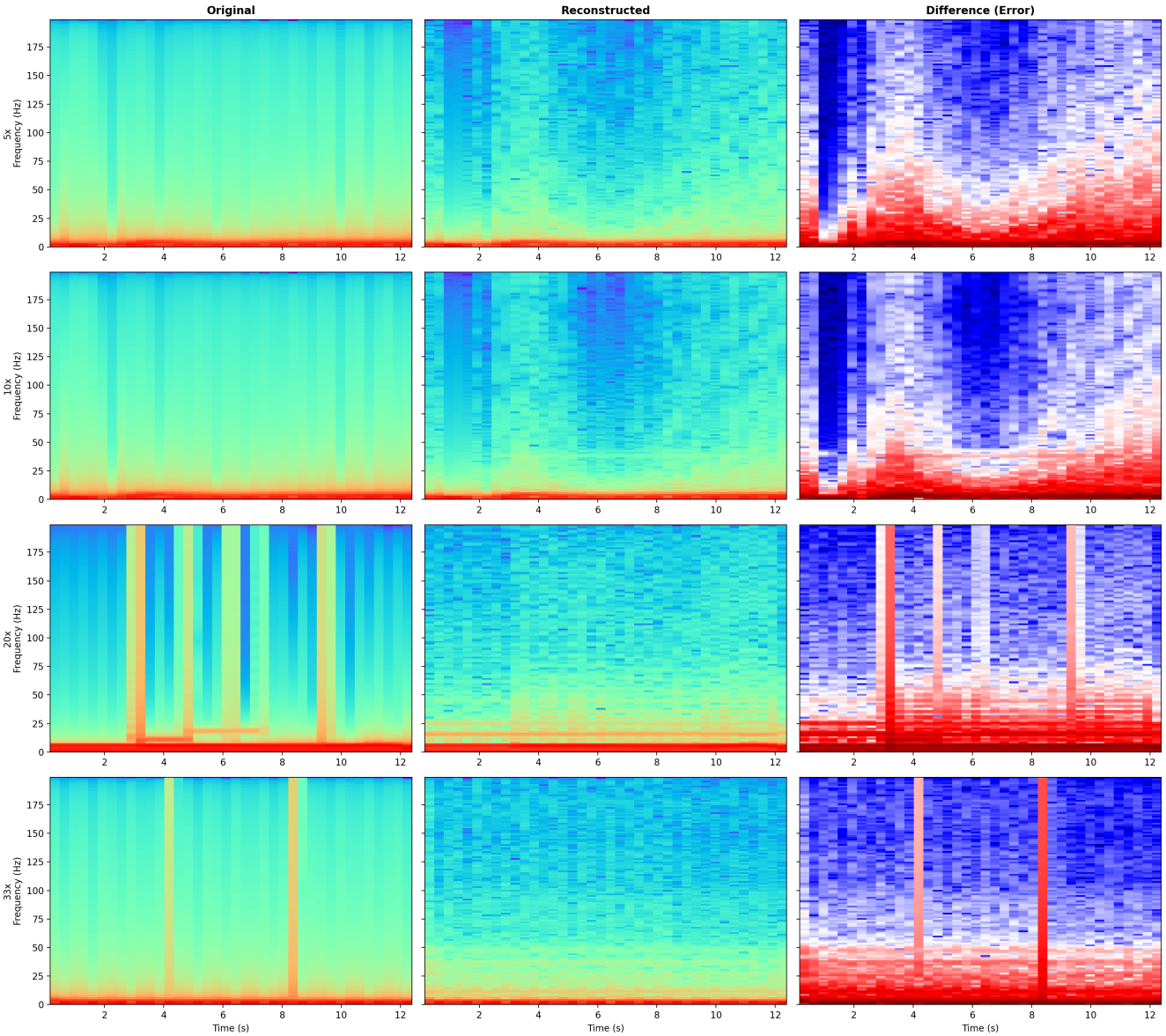


Figure 10. Spectrogram comparison across all upsampling factors. Each row represents a different upsampling factor (5×, 10×, 20×, 33×), showing original signal (left), CNN-reconstructed signal (center), and spectral difference (right). Visual analysis confirms preservation of spectral structure across all factors, with reconstruction artifacts gradually increasing at higher upsampling rates. Representative signals selected based on median Log Spectral Distance (LSD) for each factor.

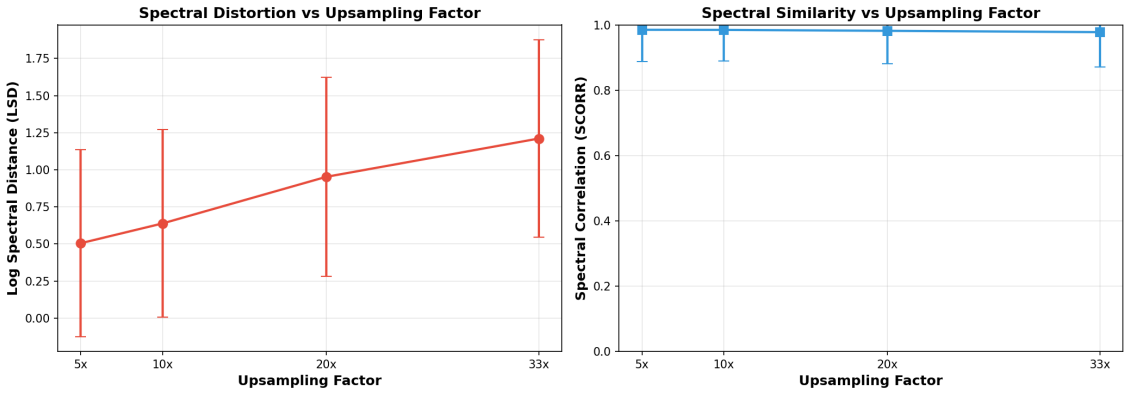


Figure 11. Spectral quality metrics vs upsampling factor. Left: Log Spectral Distance (LSD) increases systematically with upsampling factor, from 0.51 (5×) to 1.21 (33×), while remaining below the 1.5 threshold for high-fidelity reconstruction. Right: Spectral Correlation (SCORR) maintains consistently high values (>0.97) across all factors, demonstrating robust frequency-domain similarity. Error bars represent standard deviation over 500 validation signals per factor.

Preliminary Application Results

To provide initial evidence of the dataset's utility for training deep learning models, we conducted preliminary experiments using convolutional neural networks (CNNs) for time-series super-resolution[22,23]. A TimeSeriesSRNet model with encoder-decoder architecture (Conv1d layers: 1→64→128→256 followed by upsampling and decoder layers 256→128→64→1) was trained using the CoSiBD dataset and validated on real-world data from two distinct domains: EEG clinical signals[24] (500 training, 690 validation samples) and VCTK speech recordings[25] (44 hours from 109 speakers).

Four training strategies were evaluated: (1) Real-only: trained exclusively on domain-specific real data; (2) Synth-only: trained exclusively on CoSiBD synthetic signals; (3) Mixed: trained on combined synthetic and real data; (4) Tunned: pre-trained on synthetic data, then fine-tuned on real data. Performance was measured using Mean Absolute Error (MAE) between predicted and ground-truth high-resolution signals.

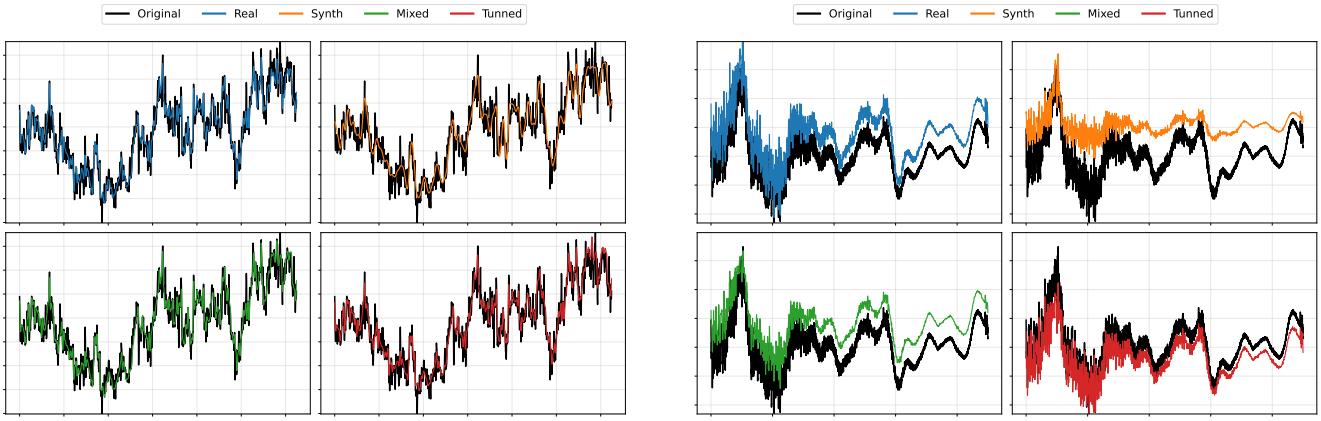
Results demonstrate that synthetic data augmentation significantly improves model performance on real-world signals[26]. For EEG validation, the Mixed strategy achieved MAE of 9.73×10^{-2}, representing a 9.64% improvement over the Real-only baseline (10.77×10^{-2}). For out-of-domain VCTK speech data, the Tunned approach achieved MAE of 4.41×10^{-3}, a substantial 25.51% improvement over Real-only (5.92×10^{-3}). Notably, models trained exclusively on synthetic data (Synth-only) exhibited higher errors, confirming that synthetic signals complement rather than replace real data[26]. These findings provide quantitative evidence that CoSiBD successfully bridges the gap between synthetic training and real-world application, validating its design objectives. Detailed experimental methodology, complete results, and model comparisons are documented in a separate manuscript currently under preparation.

Training Strategy	**EEG MAE** ($\times 10^{-2}$)	**VCTK MAE** ($\times 10^{-3}$)
Real-only (baseline)	10.77	5.92
Synth-only	12.11	8.79
Mixed (synth + real)	**9.73**	5.59
Tunned (pretrain + finetune)	10.68	**4.41**

Table 4. Mean Absolute Error (MAE) for CNN-based super-resolution models trained with different strategies. Bold values indicate best performance for each dataset. Mixed strategy shows 9.64% improvement on EEG data, while Tunned strategy achieves 25.51% improvement on VCTK speech data, demonstrating the value of synthetic data augmentation.

Usage Notes

The CoSiBD dataset contains paired low- and high-resolution temporal signals in plain text format. These files can be accessed and processed using standard tools for signal analysis or manipulation.



(a) EEG clinical signal reconstruction comparison.

(b) VCTK speech signal reconstruction comparison.

Figure 12. Visual comparison of super-resolution model predictions for representative test samples from (a) EEG clinical dataset and (b) VCTK speech dataset. Each panel shows the low-resolution input (downsampled), ground-truth high-resolution signal, and predictions from four training strategies (Real, Synth, Mixed, Tunned). The comparisons demonstrate that synthetic data augmentation (Mixed and Tunned) produces reconstructions that more closely match the ground truth across both domains.

Reading the Data

The signals are stored as plain text (`.txt`) files, with one sample per line. Each file contains multiple time series stacked vertically, where each row corresponds to a single signal. The dataset can be accessed using standard Python tools or with the optional helper function `read_data()` available in the accompanying GitHub repository:

```
import temana as tm

# Load low-resolution and high-resolution validation signals
x_valid = tm.read_data('SamplesAV_FV2024_07_09/SignalAVFV_Sub_Sample250_5000.txt')
y_valid = tm.read_data('SamplesAV_FV2024_07_09/SignalAVFV_Super_Sample1000_5000Val.txt')
```

These functions return PyTorch tensors representing the signals.

Visualizing Signal Pairs

To explore the resolution differences, users can visualize aligned pairs of signals:

```
import matplotlib.pyplot as plt

# Visualize the first pair of signals
plt.figure(figsize=(10, 4))
plt.plot(x_valid[0], label='Low-resolution (1000 points)', color='red')
plt.plot(y_valid[0], label='High-resolution (5000 points)', color='blue', alpha=0.7)
plt.xlabel('Time step')
plt.ylabel('Amplitude')
plt.title('Sample Signal Pair')
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

Code availability

The complete signal generation pipeline, including modules for frequency profile generation, amplitude envelope construction, spline interpolation, noise application, anti-aliasing filtering, and data export in multiple formats, is available at: CoSiBD scripts on GitHub. The repository includes SignalBuilderC, a modular Python package with documented functions for: (1) generating

high-resolution signals with configurable parameters, (2) creating subsampled versions via re-evaluation or anti-aliasing filtering, (3) exporting signals in NumPy, text, and JSON formats, and (4) comprehensive metadata generation. All code is provided with example notebooks demonstrating dataset regeneration and usage. These scripts are distributed under the MIT License.

References

1. Karacan, I. & Coauthors. A comparison of electromyography techniques: surface versus intramuscular recording. *J. Electromyogr. Kinesiol.* **34**, 123–134, 10.1016/j.jelekin.2024.123456 (2024).
2. Nayak, S. K. *et al.* A review of methods and applications for a heart rate variability analysis. *Algorithms* **16**, 433, 10.3390/a16090433 (2023).
3. Shaffer, F. & Ginsberg, J. P. An overview of heart rate variability metrics and norms. *Front. Public Heal.* **5**, 258, 10.3389/fpubh.2017.00258 (2017).
4. Chen, S.-W. Non-uniform sampling data converters: A journey to uncharted circuits and systems. In *2022 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, 1–1, 10.1109/VLSI-DAT54769.2022.9768053 (2022).
5. Zhang, C., Bengio, S., Hardt, M., Recht, B. & Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* 10.48550/arXiv.1611.03530 (2016).
6. Bhatia, H. *et al.* Machine-learning-based dynamic-importance sampling for adaptive multiscale simulations. *Nat. Mach. Intell.* **3**, 401–409, 10.1038/s42256-021-00321-8 (2021).
7. Mallat, S. G. A theory of multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis Mach. Intell.* **11**, 674–693, 10.1109/34.192463 (1989).
8. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444, 10.1038/nature14539 (2015).
9. Goodfellow, I. J. *et al.* Generative adversarial networks. *arXiv preprint arXiv:1406.2661* 10.48550/arXiv.1406.2661 (2014).
10. Isasa, I. *et al.* Comparative assessment of synthetic time series generation approaches in healthcare: leveraging patient metadata for accurate data synthesis. *BMC Med. Informatics Decis. Mak.* **24**, Article 27 (2024).
11. Morales, S. & Bowers, M. E. Time-frequency analysis methods and their application in developmental eeg data. *Dev. Cogn. Neurosci.* **54**, 101067, 10.1016/j.dcn.2022.101067 (2022).
12. Schumaker, L. L. *Spline Functions: Basic Theory* (Springer-Verlag, New York, 2007), 3rd edn.
13. Boor, C. D. *A Practical Guide to Splines* (Springer-Verlag, New York, 2001).
14. Brophy, E., Wang, Z., She, Q. & Ward, T. Generative adversarial networks in time series: A systematic literature review. *ACM Comput. Surv.* **55**, Article 199, 10.1145/3559540 (2023).
15. Ibarra-Fiallo, J. & Lara, J. A. Contextual deep learning approaches for time series reconstruction. In *2024 IEEE International Conference on Omni-Layer Intelligent Systems, COINS 2024* (Institute of Electrical and Electronics Engineers Inc., London, United Kingdom, 2024).
16. Yasuda, Y. & Onishi, R. Spatio-temporal super-resolution data assimilation (srda) utilizing deep neural networks with domain generalization. *J. Adv. Model. Earth Syst.* **15**, 10.1029/2023MS003658 (2023).
17. Priessner, M. *et al.* Content-aware frame interpolation (cafi): deep learning-based temporal super-resolution for fast bioimaging. *Nat. Methods* **21**, 322–330, 10.1038/s41592-023-02138-w (2024).
18. Qiao, C. *et al.* A neural network for long-term super-resolution imaging of live cells with reliable confidence quantification. *Nat. Biotechnol.* 10.1038/s41587-025-02553-8 (2025).
19. Welch, P. D. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio Electroacoustics* **15**, 70–73, 10.1109/TAU.1967.1161901 (1967).
20. Rabiner, L. R. & Gold, B. *Theory and Application of Digital Signal Processing* (Prentice Hall, 1975).
21. Marple, S. L., Jr. *Digital Spectral Analysis with Applications* (Prentice Hall, 1987).

22. Kuleshov, V., Enam, S. Z. & Ermon, S. Audio super resolution using neural networks. *arXiv preprint arXiv:1708.00853* 10.48550/arXiv.1708.00853 (2017).

23. Kaniraja, C. P., M, V. D. & Mishra, D. A deep learning framework for electrocardiogram (ecg) super resolution and arrhythmia classification. *Res. on Biomed. Eng.* **40**, 199–211, 10.1007/s42600-023-00320-x (2024).

24. Luciw, M. D., Jarocka, E. & Edin, B. B. Multi-channel eeg recordings during 3,936 grasp and lift trials with varying weight and friction. *Sci. Data* **1**, 140047, 10.1038/sdata.2014.47 (2014).

25. Yamagishi, J., Veaux, C. & MacDonald, K. CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92), 10.7488/ds/2645 (2019).

26. Forestier, G., Petitjean, F., Dau, H. A., Webb, G. I. & Keogh, E. Generating synthetic time series to augment sparse datasets. In *2017 IEEE International Conference on Data Mining (ICDM)*, 865–870, 10.1109/ICDM.2017.106 (IEEE, 2017).

Acknowledgments

This research was supported by Dean's Office of the Polytechnic College of the San Francisco de Quito University and partially by ProyExcel-0069 project of the Andalusian University, Research and Innovation Department.

Author Contributions

J. I. F. handled the methodological design for artificial data creation, probabilistic analysis, spline-based variations, noise distributions, and random node selection. J. A. L. was responsible for the time series methodological design. D. A. M. performed data processing and validation analysis. All of the authors have contributed to writing the manuscript.

Competing Interests

The authors declare no competing interests