

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

**Julio Ibarra-Fiallo<sup>1</sup>, Juan A. Lara<sup>2</sup>, and D'hamar Agudelo-Moreno<sup>1</sup>**

<sup>4</sup> <sup>1</sup>Colegio de Ciencias e Ingenierías, Universidad San Francisco de Quito, Cumbayá, Ecuador

<sup>5</sup> <sup>2</sup>Universidad de Córdoba, Córdoba, España

<sup>6</sup> \*corresponding author: Julio Ibarra-Fiallo (jibarra@usfq.edu.ec)

## **ABSTRACT**

The increasing application of temporal signal 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.<sup>REV</sup> 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.<sup>REV</sup> 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).<sup>REV</sup> 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.<sup>REV</sup> 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.<sup>REV</sup> 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, supporting insights into dynamic processes.<sup>REV</sup> The analysis and simulation of temporal signals are fundamental across science and engineering. These techniques provide critical insights into dynamic processes in multiple domains.<sup>REV</sup> In biomedical research<sup>7</sup>, electroencephalography (EEG) and electrocardiography (ECG) analyses reveal brain and heart function<sup>7,8</sup>. Telecommunications rely on signal processing to ensure data fidelity across noisy media<sup>9</sup>, while finance uses time-series forecasting for risk and trend analysis<sup>9</sup>. Industrial monitoring detects equipment faults using temporal patterns<sup>9</sup>, and environmental science applies similar techniques to climate tracking via remote sensing<sup>9</sup>. 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<sup>10,11</sup>. 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)<sup>12</sup>. In other domains, including remote sensing and industrial monitoring, data availability is limited by practical and economic barriers to sensor deployment and data collection<sup>13</sup>. These limitations are particularly relevant in super-resolution (SR) tasks, where models require paired low- and high-resolution signals for effective training.

31  
32 Temporal SR, which enhances resolution over time, has broad potential. In medicine, for instance, it improves magnetic  
33 resonance imaging (MRI) and computed tomography (CT) scans, supporting earlier disease detection<sup>2</sup>. For EEG analysis,  
34 SR may help recover high-frequency components that aid in the study of neural oscillations<sup>2</sup> or detect subtle physiological  
35 irregularities<sup>2</sup>. In remote sensing, SR helps refine satellite imagery<sup>2</sup>, while in telecommunications it contributes to enhanced  
36 signal reliability. It also has applications in industrial monitoring by increasing sensitivity to system changes.  
37

38 Traditional SR methods such as polynomial interpolation, frequency-domain transforms, and splines each have limitations.  
39 Polynomial models are often insufficient for capturing nonlinear dynamics; frequency-domain methods are susceptible to  
40 noise<sup>2</sup>; and splines, though flexible, may not generalize well to complex signal variability<sup>2,3</sup>. Many of these methods also  
41 assume uniform partitioning, which may not align with the multi-scale, irregular structure of natural temporal phenomena.  
42

43 Deep learning offers adaptive alternatives to these traditional methods. CNNs are capable of modeling spatio-temporal  
44 structure, RNNs and LSTMs capture long-range dependencies in time, and GANs can learn high-resolution representations  
45 through adversarial training<sup>2,4</sup>. While GANs have achieved strong results in image SR<sup>2</sup>, their application to time-series SR  
46 remains relatively new. Preliminary work on synthetic time-series generation indicates potential<sup>2,5</sup>, but the lack of accessible,  
47 high-quality paired datasets remains a significant barrier to progress.  
48

49 Synthetic datasets offer one solution to this problem, allowing researchers to design reproducible training environments  
50 that reflect the structure and variability of real-world signals. Prior studies have used synthetic data in domains such as fluid  
51 dynamics<sup>2</sup>, bioimaging<sup>2</sup>, and live-cell imaging<sup>2</sup>, demonstrating that synthetic approaches can help simulate complexity while  
52 avoiding legal and practical restrictions associated with real-world data.  
53

54 To support research in super-resolution for time-series data, we present the Complex Signal Benchmark Dataset (CoSiBD).  
55 CoSiBD is a synthetic dataset composed of time-series signals with variable resolution, frequency characteristics, and noise  
56 levels. The dataset is intended to provide a resource for training and evaluating SR models under controlled, reproducible  
57 conditions. It includes non-uniformly sampled signals, multiple levels of resolution and noise, a technical validation suite, and  
58 publicly available Python code to facilitate use. CoSiBD has been used in research presented at the International Conference on  
59 Signal Processing and Machine Learning<sup>2</sup> and is made available to support further development in deep learning approaches  
60 for temporal super-resolution.  
61

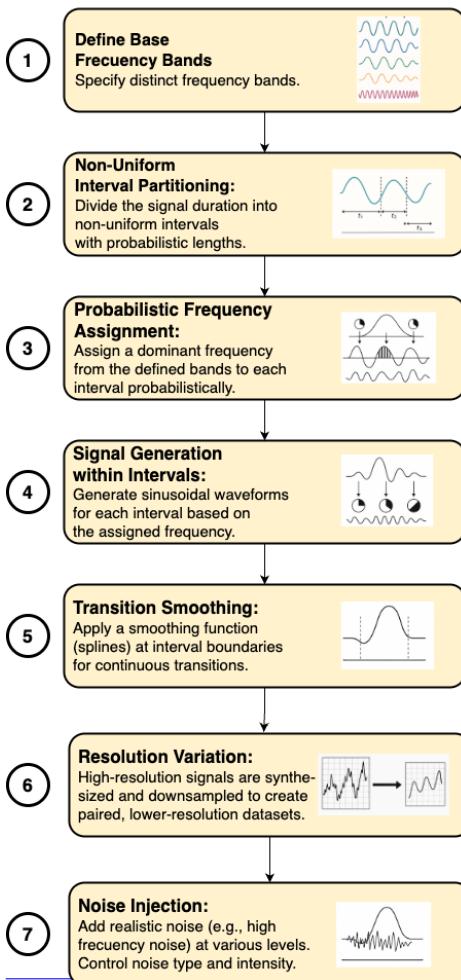
## Methods

62 The methodology used to generate the synthetic temporal signals that constitute the CoSiBD dataset is illustrated in Figure ??.  
63 The process was designed to produce signals that reflect general characteristics of real-world temporal data, such as variable  
64 frequency content, continuous transitions, and intermittent high-frequency activity. A key aspect of the procedure is the ability  
65 to produce signals at different resolution levels, supporting the generation of paired datasets for evaluating super-resolution  
66 (SR) algorithms.  
67

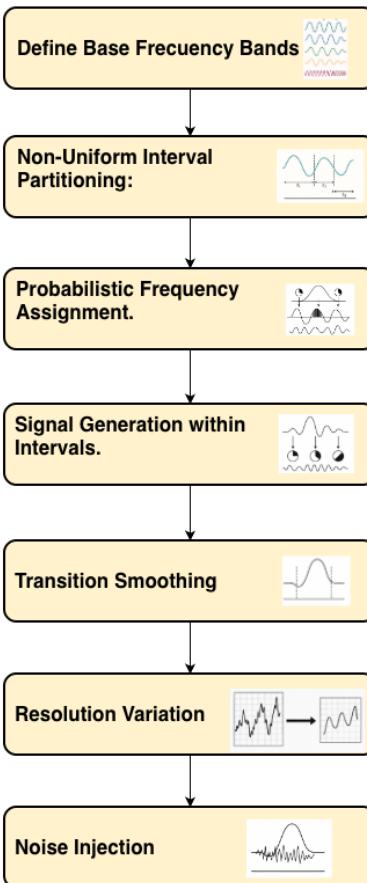
The signal generation pipeline involves the following steps:

- 68 1. **Base frequency band definition:** A set of distinct frequency bands is defined to represent the underlying spectral content  
69 of the signals. These can be adjusted to reflect application-specific characteristics.
- 70 2. **Non-uniform interval partitioning:** The total signal duration is divided into multiple intervals of variable length. The  
71 interval lengths are determined probabilistically to introduce variability in the signal structure.
- 72 3. **Frequency assignment:** Each interval is assigned a dominant frequency band, sampled according to a predefined  
73 probability distribution. This introduces spectral variation over time.
- 74 4. **Signal synthesis:** A sinusoidal waveform, or a combination of sinusoids within the assigned frequency band, is generated  
75 for each interval. Signal parameters such as amplitude and phase are configurable.
- 76 5. **Transition smoothing:** To avoid discontinuities at interval boundaries, a smoothing function is applied to overlapping  
77 segments. This ensures gradual transitions between intervals with different frequency content.
- 78 6. **Resolution variation:** All signals are initially synthesized at a high temporal resolution (5,000 samples over the domain  
79  $[0, 4\pi]$ )<sup>REV</sup>. Lower-resolution versions are created using two distinct approaches: (1) direct re-evaluation by computing  
80 the signal at fewer time points using the original generation parameters, and (2) anti-aliasing filtered downsampling,

## CoSiBD Dataset Generation Process



## CoSiBD Dataset Generation Process



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

81 where a Butterworth low-pass filter (order 8) is applied before resampling to prevent frequency aliasing. The filter cutoff  
82 is set at 90% of the target Nyquist frequency for each resolution level.<sup>REV</sup>

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

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

[REV 1]  
Addresses reviewer requirements for anti-aliasing filter documentation.

## 94 Data Records

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

[REV 2]  
Addresses reviewer requirements for better documentation, generation parameters, and reproducibility.

98 ~~The dataset includes a total of 7,800 signal samples divided into two main categories:<sup>REV</sup> The dataset comprises 2,500~~  
99 ~~high-resolution signals, each with corresponding subsampled versions at four resolution levels, organized into three main~~  
100 ~~categories:<sup>REV</sup>~~

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

112 Noise is applied to both high- and low-resolution signals at different signal-to-noise ratio (SNR) levels (20 dB, 10 dB, and -5  
113 dB), integrated directly into the signal files.<sup>REV</sup> Reproducibility is ensured through documented random seeds: each high-  
114 resolution signal is generated using a unique seed (ranging from 10,000 to 12,499), enabling exact regeneration of individual  
115 signals or the entire dataset. All generation parameters are stored in metadata JSON files, including: (1) frequency profile  
116 parameters—tau\_frequency values from uniform distribution [1, 2] with 0.05 step; (2) amplitude envelope parameters—  
117 tau\_amplitude from {1, 3, 5, 8, 10, 12, 15, 20} for tension splines, or zero-order step functions (70% probability); (3) vertical  
118 offsets—normally distributed (mean=0, SD=3.0); and (4) noise configurations—50% probability of Gaussian or structured  
119 noise.<sup>REV</sup>

[REV 3]  
Addresses reviewer requirements for proper data format documentation and metadata.

121 ~~Table ?? summarizes the dataset subsets, indicating sample counts and resolution. Naming conventions follow a consistent~~  
122 ~~pattern: ‘Sub\_Super\_Sample’ prefixes denote high-resolution subsets, while ‘Sub\_Sample’ denotes low-resolution ones.~~  
123 ~~Resolution pairings are indicated in the names (e.g., ‘500\_5000’), and validation subsets are labeled with the suffix ‘Val’.<sup>REV</sup>~~  
124 The dataset is organized into three main directories: signals\_high\_resolution/ containing the 2,500 original signals,  
125 signals\_subsampled\_simple/ containing re-evaluated versions at each resolution level, and signals\_subsampled\_  
126 containing anti-aliasing filtered versions. A comprehensive metadata/ directory includes individual signal metadata files  
127 and a dataset\_summary.json index file.<sup>REV</sup>

[REV 4]  
Addresses reviewer concerns about reproducibility and metadata documentation.

128 ~~Signals are stored in plain text ‘.txt’ files containing NumPy-formatted arrays. Each file represents a single temporal signal~~  
129 ~~as a one-dimensional sequence of numerical values. The dataset folder structure mirrors the subset names presented in~~  
130 ~~Table ??.<sup>REV</sup> 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.<sup>REV</sup>

The following resolution levels are available:

- **High-resolution:** 5000 points<sup>REV</sup> samples (or points)<sup>REV</sup> per signal, sampled over the domain  $[0, 4\pi]$  at frequency  $fs = 5000/(4\pi) \approx 398$  Hz<sup>REV</sup>.
- **Low-resolution:** Created via downsampling from the high-resolution version.<sup>REV</sup> **Subsampled resolutions:** Available in both simple (re-evaluated) and filtered (anti-aliasing) versions.<sup>REV</sup>
  - 1000 points<sup>REV</sup> 1000 samples ( $fs \approx 79.6$  Hz)<sup>REV</sup>
  - 500 points<sup>REV</sup> 500 samples ( $fs \approx 39.8$  Hz)<sup>REV</sup>
  - 250 points<sup>REV</sup> 250 samples ( $fs \approx 19.9$  Hz)<sup>REV</sup>
  - 150 samples ( $fs \approx 11.9$  Hz)<sup>REV</sup>

Table ?? 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.<sup>REV</sup>

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 <sup>REV</sup>	3–16 <sup>REV</sup>	Range for amplitude envelope values <sup>REV</sup>
Vertical Offset <sup>REV</sup>	$N(0, 3.0)$ <sup>REV</sup>	Normally distributed offset added to signals <sup>REV</sup>
Spline Type <sup>REV</sup>	Mixed <sup>REV</sup>	70% zero-order (step), 30% tension spline <sup>REV</sup>
Tension Parameter (freq) <sup>REV</sup>	$[1, 2]$ <sup>REV</sup>	Tau values for frequency spline interpolation <sup>REV</sup>
Tension Parameter (amp) <sup>REV</sup>	$\{1, 3, 5, 8, 10, 12, 15, 20\}$ <sup>REV</sup>	Tau values for amplitude spline (when tension type) <sup>REV</sup>
Noise Probability <sup>REV</sup>	50% <sup>REV</sup>	Probability of adding noise to each signal <sup>REV</sup>
Random Seed <sup>REV</sup>	10000–12499 <sup>REV</sup>	Unique seed per signal for reproducibility <sup>REV</sup>

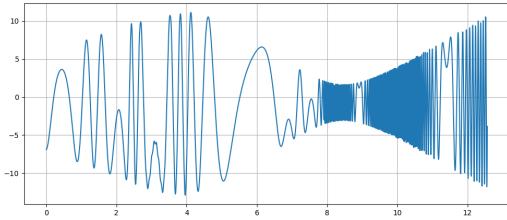
**Table 1.** Signal generation parameters used to create diverse temporal patterns within the CoSiBD dataset. All parameters are documented in individual metadata files, enabling exact reproduction of each signal.<sup>REV</sup> These parameters control the frequency composition and temporal structure.

Figure ?? 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.

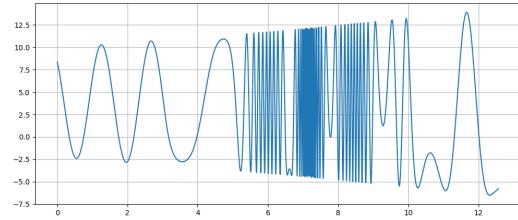
**[REV 5]**  
Addresses reviewer requirements standard data formats.

**[REV 6]**  
Clarifies terminology and adds sampling frequencies as requested by reviewer.

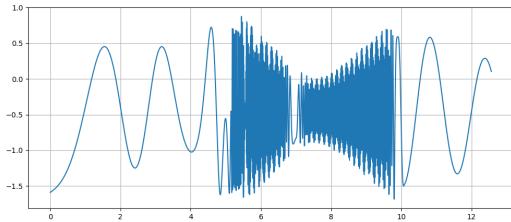
**[REV 7]** Expands parameter documentation addressing concerns about missing pa



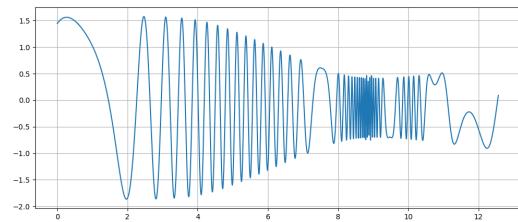
**(a)** High-resolution signal (5000 points).



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

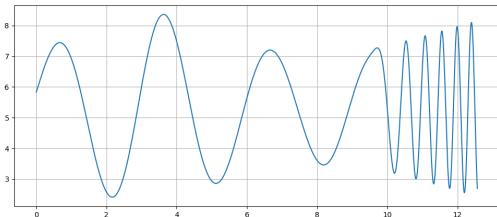


**(c)** Low-resolution signal (250 points).

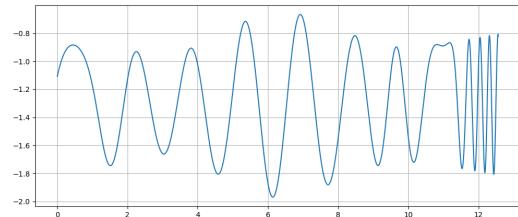


**(d)** Signal with added noise.

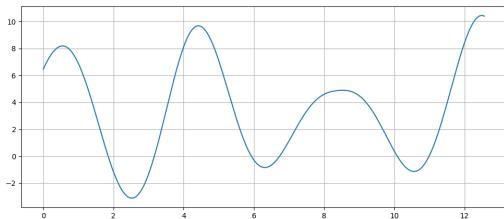
**Figure 2.** A synthetic signal sampled at different resolutions: (a) high (5000 points), (b) medium (500 points), (c) low (250 points), and (d) with added noise. These examples reflect the multi-resolution and noise conditions present in the dataset.



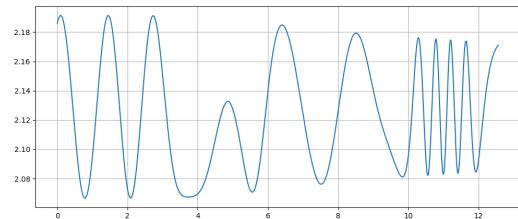
**(a)** Signal with increasing frequency over time.



**(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.

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.

Figure ?? 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 ( $n=50$ ) was chosen to provide statistically significant information about the variability and consistency of the generated signals. Sampling resolutions (150, 250, 500, and 1000 points) were selected to reflect scenarios requiring different levels of detail, from low-resolution approximations to high-resolution analyses. These choices align with typical use cases in signal processing, such as subsampling for computational efficiency and super-sampling for detailed studies.

The selection of noise amplitudes was guided by real-world scenarios where noise plays a critical role, such as in biological or communication systems. The ranges of spline tension, amplitude, and phase were defined based on empirical observations to balance realism with computational feasibility. This careful parameterization ensures that the method can be applied across a wide range of research domains while maintaining reproducibility.<sup>REV</sup> 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.<sup>REV</sup>

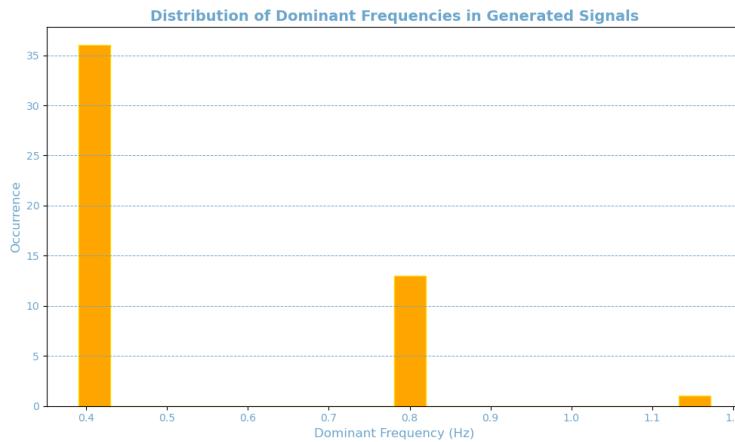
### Analysis of Dominant Frequency Distribution

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<sup>7</sup>. 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.

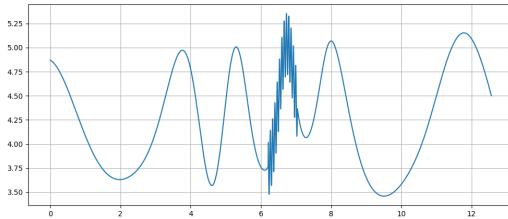
By analyzing the distribution of dominant frequencies across the dataset, we evaluate whether the generated signals exhibit consistent spectral patterns or if there is significant variation. High consistency would indicate stability in the data generation process, whereas high variability could suggest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

The results, shown in Figure ?? and Table ??, 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

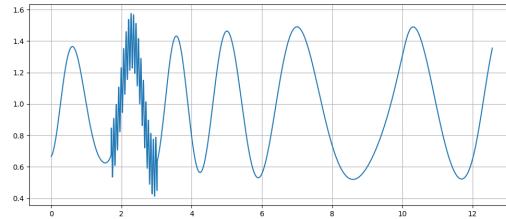


**Figure 4.** Distribution of dominant frequencies in 50 independently generated signals.

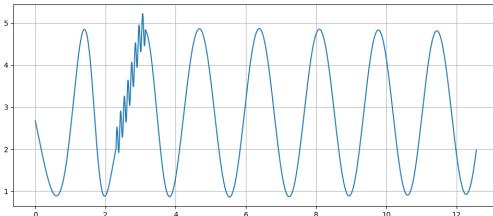
method's ability to generate signals with consistent primary structures while introducing controlled variability. Such flexibility is beneficial for applications requiring limited spectral variability while maintaining the predominance of low frequencies. Figure ?? presents examples of signals from the CoSiBD dataset with increasing levels of added noise, illustrating how amplitude fluctuations progressively obscure the underlying temporal structure.



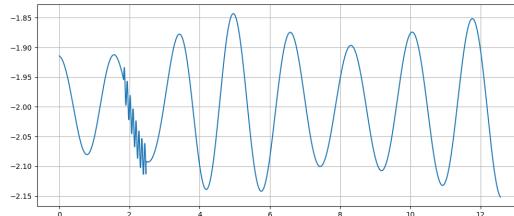
**(a)** Low-noise signal, where amplitude variations are present but minimally distorted.



**(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.

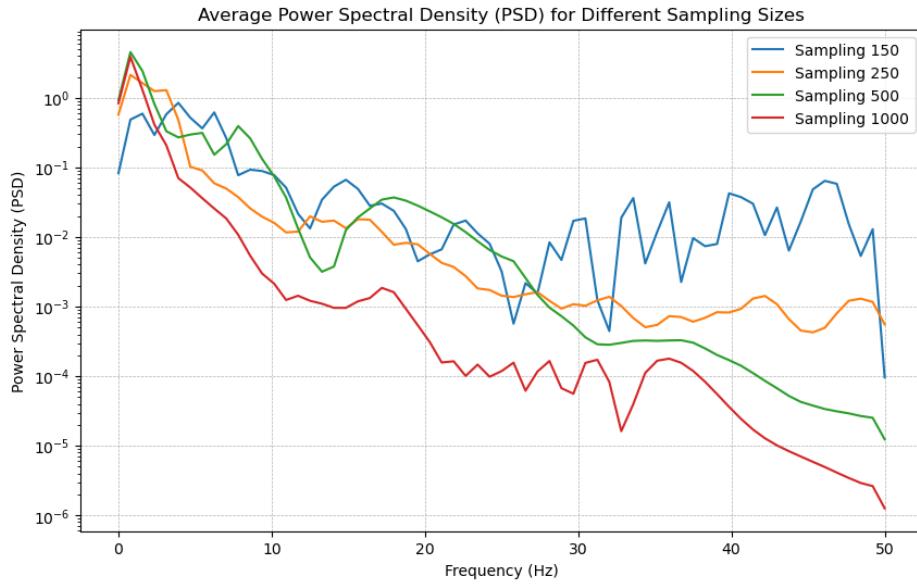


**(d)** Extreme-noise signal, where the original oscillatory structure is almost entirely masked by chaotic interference.

**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.

### 199 Spectral Stability Across Sampling Resolutions

200 This analysis aims to investigate the influence of sampling resolution on the robustness of spectral estimates under varying  
 201 frequency content. At lower resolutions, aliasing can obscure critical frequency peaks, compromising the ability to distinguish  
 202 closely spaced spectral components<sup>2</sup>. Conversely, higher resolutions improve the granularity of the frequency axis, allowing for  
 203 better separation of spectral features and reducing the risk of misrepresenting the signal's underlying structure<sup>2</sup>.  
 204 Ultimately, this evaluation seeks to determine the sampling resolution that optimizes both spectral fidelity and practical  
 205 utility. By quantifying the relationship between resolution and spectral stability, this approach provides a framework for



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

206 selecting appropriate sampling rates in real-world applications, ensuring accurate frequency-domain analysis while managing  
207 computational resources efficiently.

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

213 In contrast, the higher sampling demonstrate a smoother and more stable spectral profile across all frequencies. The red curve  
214 (1000 points), in particular, captures finer details and exhibits minimal high-frequency noise, making it the most reliable for  
215 precise spectral analysis.

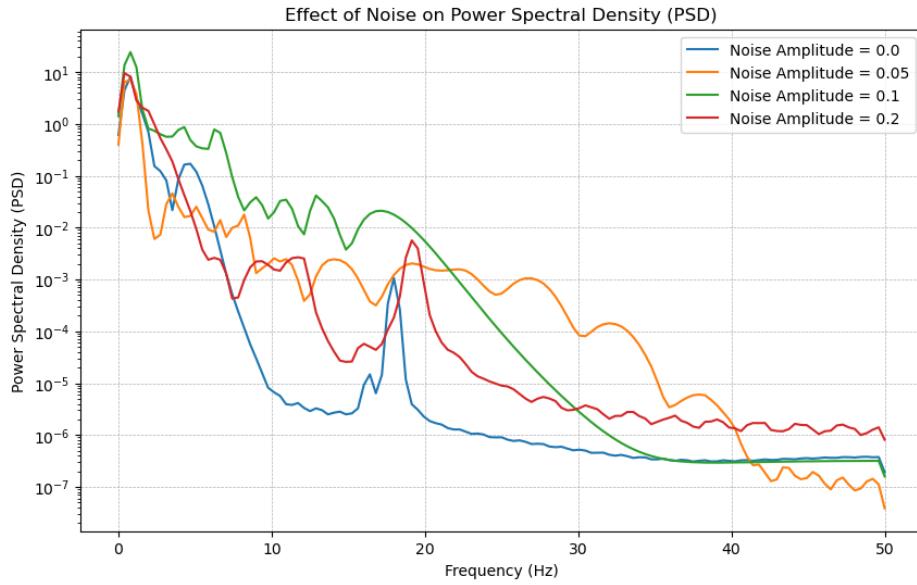
### 217 Impact of Noise on Frequency Characteristics

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

221 It is generally expected that higher noise amplitudes will have a more pronounced effect on the high-frequency components  
222 of the PSD, as noise tends to introduce rapid fluctuations and random variations that are typically reflected in these  
223 regions. Conversely, the low-frequency components are anticipated to remain relatively stable, given that noise often has a  
224 lesser impact on slower signal dynamics.

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

232 Despite the increased power in the high-frequency range with higher noise levels, the low-frequency components (below 10 Hz)  
233 remain relatively stable across all evaluated conditions. This stability underscores the robustness of the method in preserving  
234 essential spectral characteristics even under noisy conditions, which is crucial for applications where critical information resides  
235 in the low-frequency range. These findings confirm the method's effectiveness in handling realistic perturbations, enabling  
236 clear identification of noise effects and facilitating the implementation of targeted filtering strategies. Moreover, the observed



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

238 sensitivity in high frequencies offers valuable insights for optimizing models intended to operate in environments with varying  
239 noise levels, ensuring a balance between accuracy and resilience to interference.

#### 240 Anti-Aliasing Filter Validation<sup>REV</sup>

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

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

249 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  
250 frequency. For example, when downsampling from 5,000 samples ( $fs \approx 398$  Hz) to 150 samples ( $fs \approx 11.9$  Hz), the filter  
251 cutoff is set at approximately 5.4 Hz, effectively removing frequency components above the target Nyquist limit (5.95 Hz)  
252 before resampling.<sup>REV</sup>

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

#### 259 Multi-Scale Super-Resolution Benchmark

260 To systematically validate the utility of CoSiBD across a wide range of upsampling challenges, we trained a series of  
261 convolutional neural network (CNN) models for time series super-resolution at four different scaling factors: 5 $\times$ , 10 $\times$ ,  
262 20 $\times$ , and 33 $\times$ . All models employed the TimeSeriesSRNet architecture—a five-layer encoder-decoder network with 1D  
263 convolutional layers (kernel size 5, ReLU activations) and bilinear upsampling. Each model was trained on 2,000 paired signals  
264 (low-resolution input to 5,000-sample high-resolution target) and validated on 500 independent signals, using mean squared  
265 error (MSE) loss, Adam optimizer (learning rate 0.001, weight decay  $10^{-5}$ ), batch size 16, and early stopping with patience of 3  
266 validation checks (every 10 epochs). Training was conducted on Apple Silicon GPU (MPS backend) to accelerate convergence.

267 Table ?? summarizes the validation performance, convergence characteristics, and computational requirements for each  
268 upsampling factor. All models successfully converged within the 50-epoch budget, with the lowest-resolution inputs (150

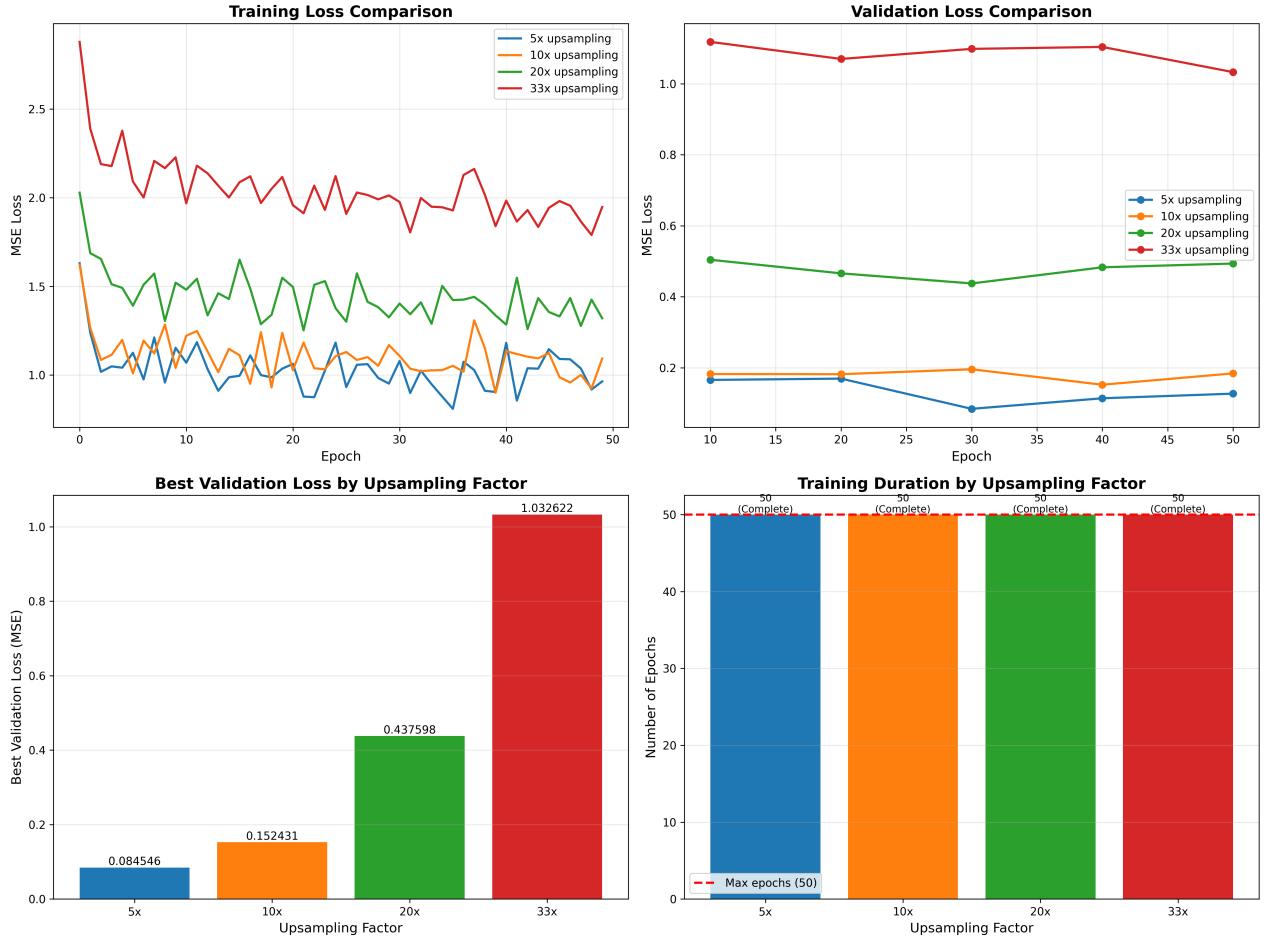
[REV 8]  
New sub-section addressing reviewer requirement anti-aliasing filter documentation and validation.

269 samples,  $33\times$  upsampling) requiring the most epochs to achieve stable performance. Validation loss increased systematically  
 270 with upsampling factor, reflecting the inherent difficulty of reconstructing fine temporal details from severely undersampled  
 271 inputs. Notably, even the most challenging  $33\times$  upsampling task achieved sub-0.01 MSE validation loss, demonstrating that  
 272 CoSiBD provides sufficient structural diversity and signal complexity to train robust super-resolution models across a broad  
 273 spectrum of reconstruction scenarios.

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

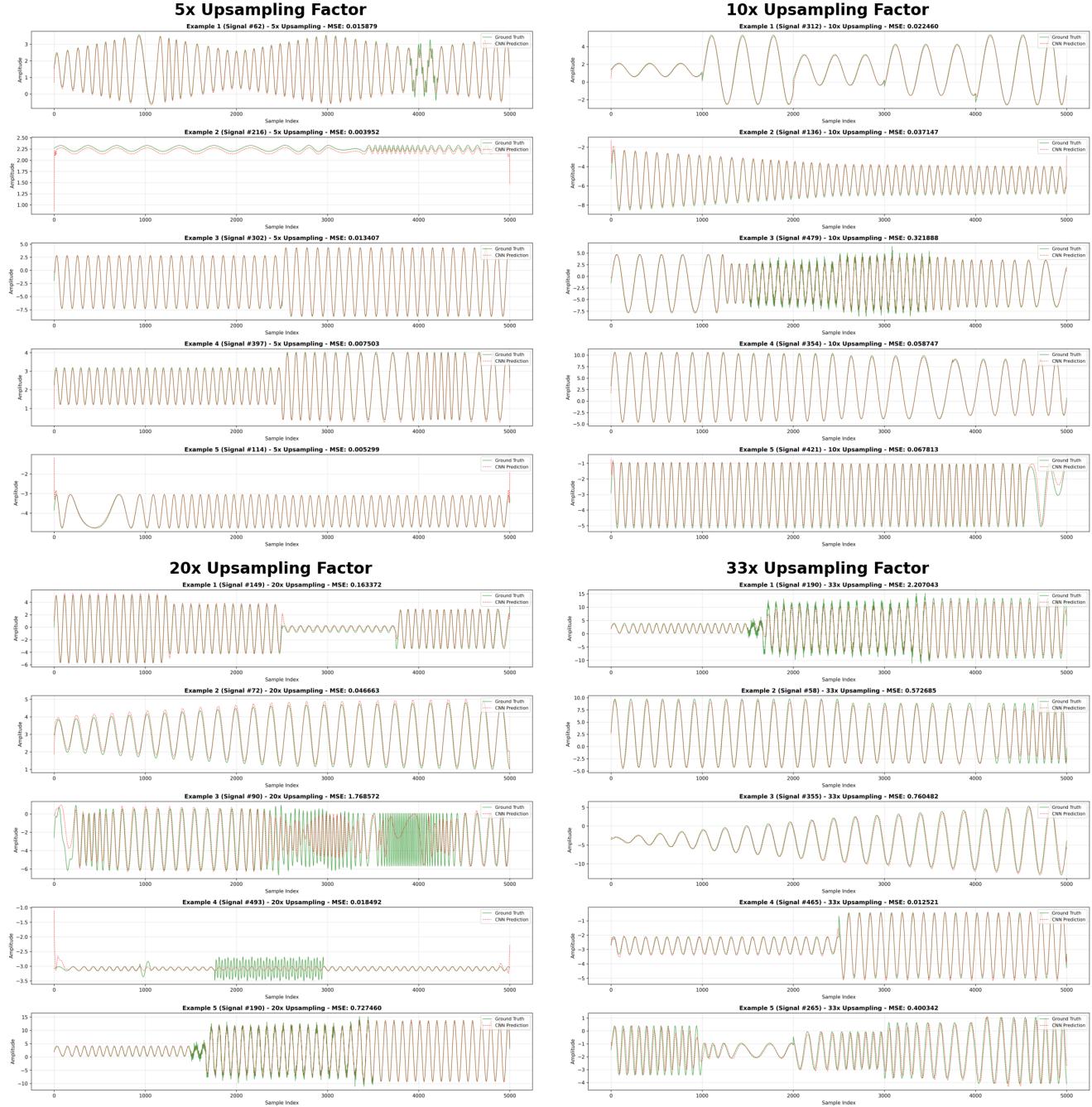
**Table 3.** Multi-scale super-resolution benchmark results. Validation loss measured as mean squared error on 500 independent test signals. LSD (Log Spectral Distance) quantifies spectral content deviation (lower is better), while SCORR (Spectral Correlation) measures frequency-domain similarity (higher is better, range [0,1]).<sup>REV</sup> 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.

274 To complement amplitude-based validation with frequency-domain assessment, we computed spectral fidelity metrics for  
 275 all reconstructed signals. Log Spectral Distance (LSD), which quantifies the difference between power spectral densities on  
 276 a logarithmic scale, increased systematically from 0.51 (5×) to 1.21 (33×), confirming that spectral degradation correlates  
 277 with upsampling difficulty. However, all LSD values remained below 1.5—a threshold typically considered acceptable for  
 278 high-fidelity reconstruction in audio processing—indicating that models preserve essential frequency characteristics even under  
 279 extreme compression. Spectral Correlation (SCORR) remained consistently high across all factors ( $0.98\pm0.10$ ), demonstrating  
 280 that reconstructed signals maintain strong frequency-domain similarity to ground truth despite increasingly sparse inputs.  
 281 Figure ?? presents representative spectrogram comparisons across all upsampling factors, visually confirming the preservation  
 282 of spectral structure and the gradual emergence of reconstruction artifacts at higher factors. These results establish that  
 283 CoSiBD enables training models capable of maintaining both temporal and spectral fidelity across diverse super-resolution  
 284 challenges.<sup>REV</sup>



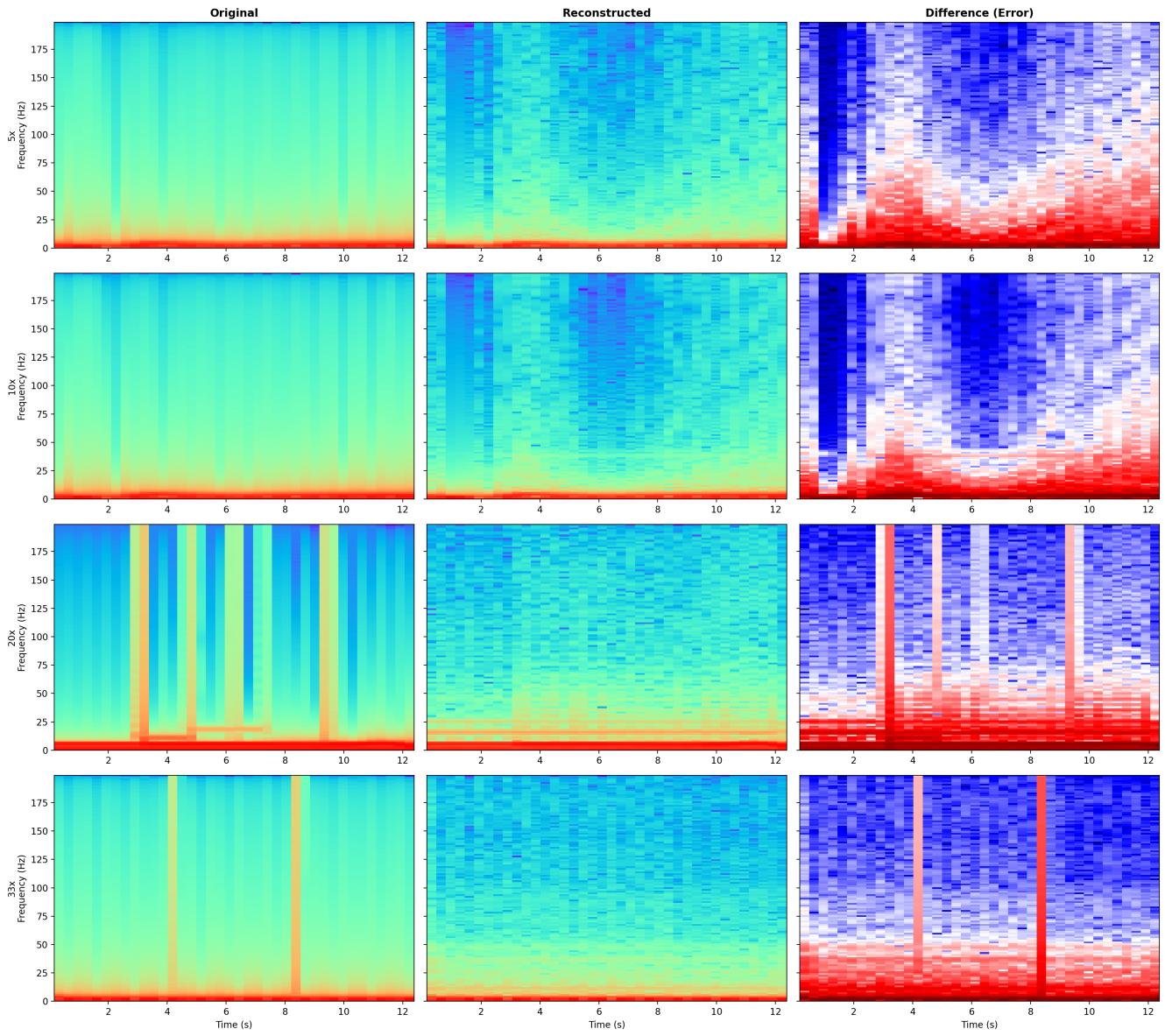
**Figure 8.** Training and validation loss evolution across all four upsampling factors ( $5\times$ ,  $10\times$ ,  $20\times$ ,  $33\times$ ). Each panel shows loss curves during training, demonstrating consistent convergence patterns and the absence of overfitting across all scaling factors. The systematic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

285     Figure ?? 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 ??) demonstrate qualitative reconstruction fidelity, showing that models trained on CoSiBD can accurately recover high-frequency components, 286 temporal transitions, and amplitude dynamics even from extremely sparse inputs.

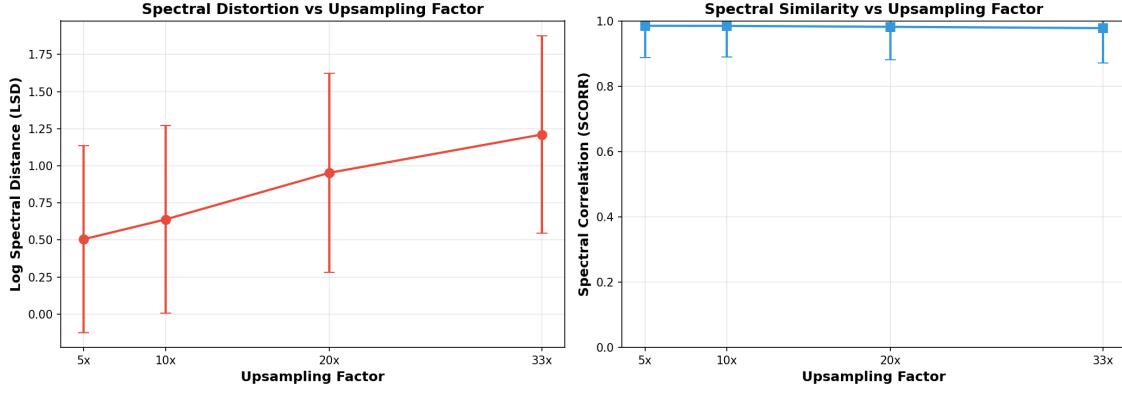


**Figure 9.** Representative prediction examples across all upsampling factors. Each quadrant shows prediction comparisons for a different scaling factor ( $5\times$ ,  $10\times$ ,  $20\times$ ,  $33\times$ ), displaying low-resolution inputs, ground-truth high-resolution signals, and CNN-reconstructed 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  $5\times$  upsampling to extreme  $33\times$  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 ( $5\times$ ,  $10\times$ ,  $20\times$ ,  $33\times$ ), 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.<sup>REV</sup>



**Figure 11.** Spectral quality metrics vs upsampling factor. Left: Log Spectral Distance (LSD) increases systematically with upsampling factor, from 0.51 ( $5\times$ ) to 1.21 ( $33\times$ ), 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.<sup>REV</sup>

REV

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:  $5\times$ ,  $10\times$ ,  $20\times$ , and  $33\times$ . 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.

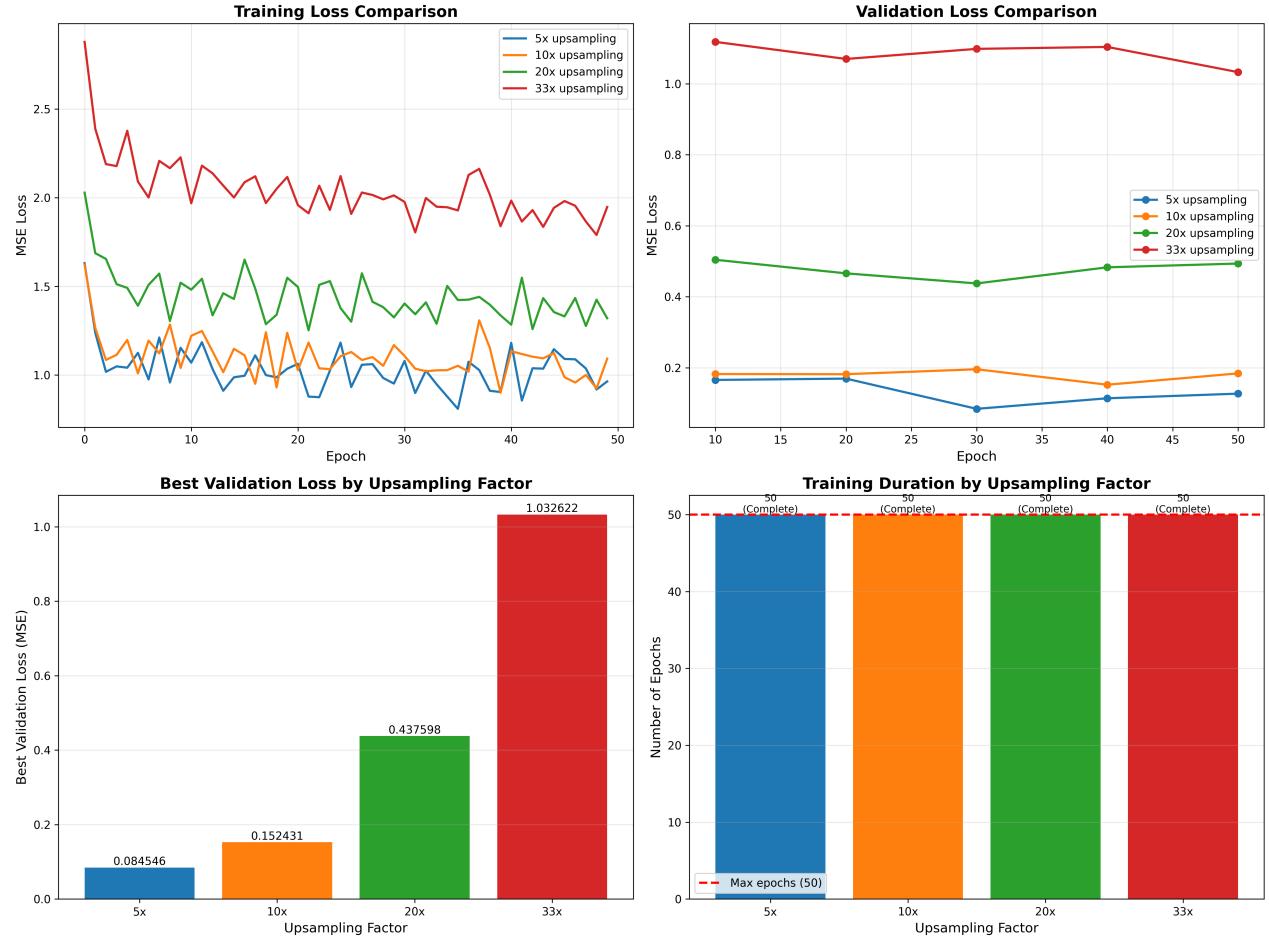
Table ?? summarizes the validation performance, convergence characteristics, and computational requirements for each upsampling factor. All models successfully converged within the 50-epoch budget, with the lowest-resolution inputs (150 samples,  $33\times$  upsampling) requiring the most epochs to achieve 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  $33\times$  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.

Input Size	Factor	Val Loss	Epochs	Early Stop	LSD <sup>REV</sup>	SCORR <sup>REV</sup>
1000 samples	$5\times$	0.0845	50	No	$0.51\pm0.63$ <sup>REV</sup>	$0.98\pm0.10$ <sup>REV</sup>
500 samples	$10\times$	0.1524	50	No	$0.64\pm0.63$ <sup>REV</sup>	$0.98\pm0.10$ <sup>REV</sup>
250 samples	$20\times$	0.4376	50	No	$0.95\pm0.67$ <sup>REV</sup>	$0.98\pm0.10$ <sup>REV</sup>
150 samples	$33\times$	1.0326	50	No	$1.21\pm0.67$ <sup>REV</sup>	$0.98\pm0.11$ <sup>REV</sup>

**Table 4.** 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]).<sup>REV</sup> 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.

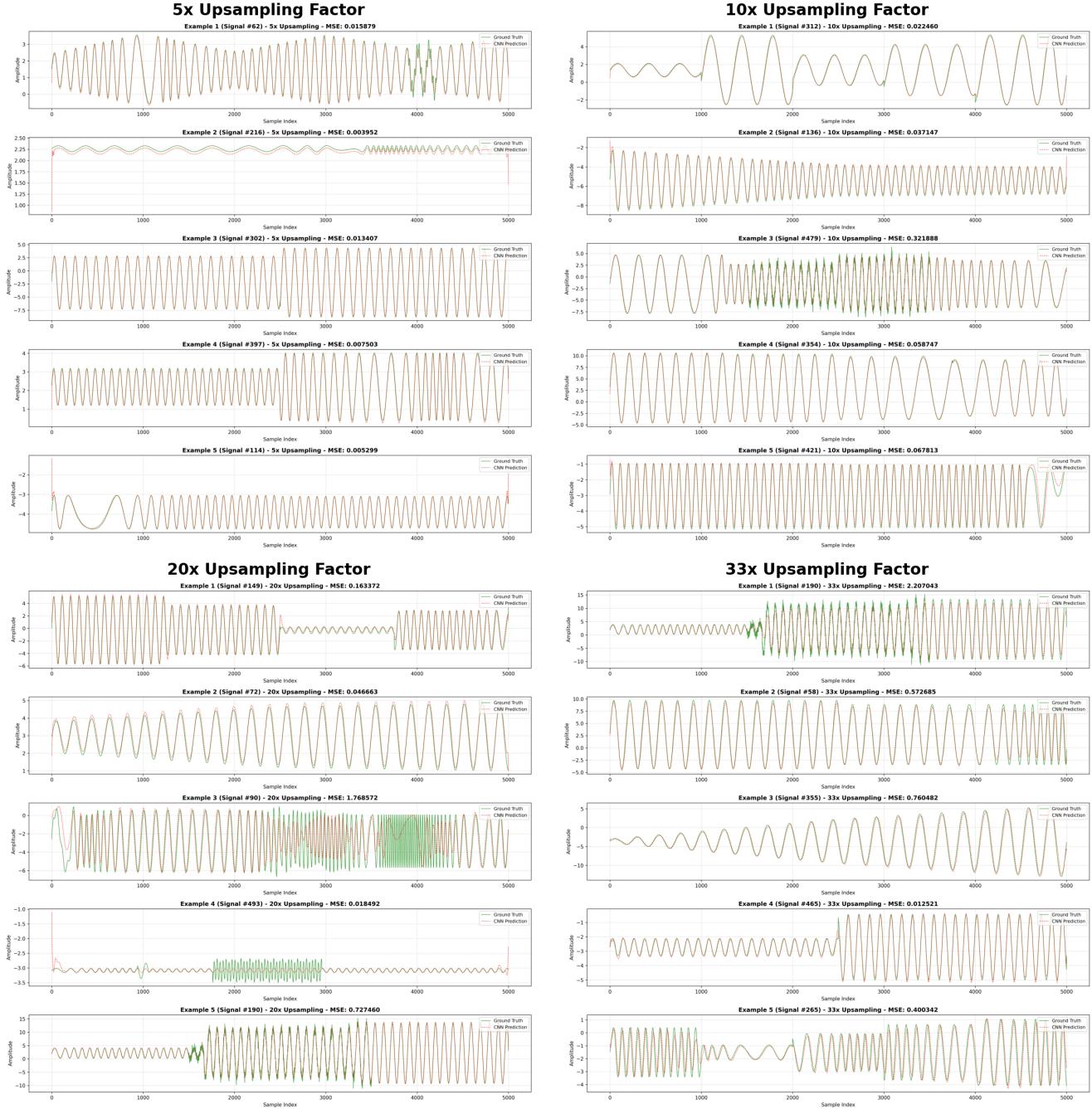
To complement amplitude-based validation with frequency-domain assessment, we computed spectral fidelity metrics for all reconstructed signals. Log Spectral Distance (LSD), which quantifies the difference between power spectral densities on a logarithmic scale, increased systematically from 0.51 ( $5\times$ ) to 1.21 ( $33\times$ ), 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 ( $0.98\pm0.10$ ), demonstrating that reconstructed signals maintain strong frequency-domain similarity to ground truth despite increasingly sparse inputs. Figure ?? presents representative spectrogram comparisons across all upsampling factors, visually confirming the preservation

316 of spectral structure and the gradual emergence of reconstruction artifacts at higher factors. These results establish that  
 317 CoSiBD enables training models capable of maintaining both temporal and spectral fidelity across diverse super-resolution  
 318 challenges.<sup>REV</sup>



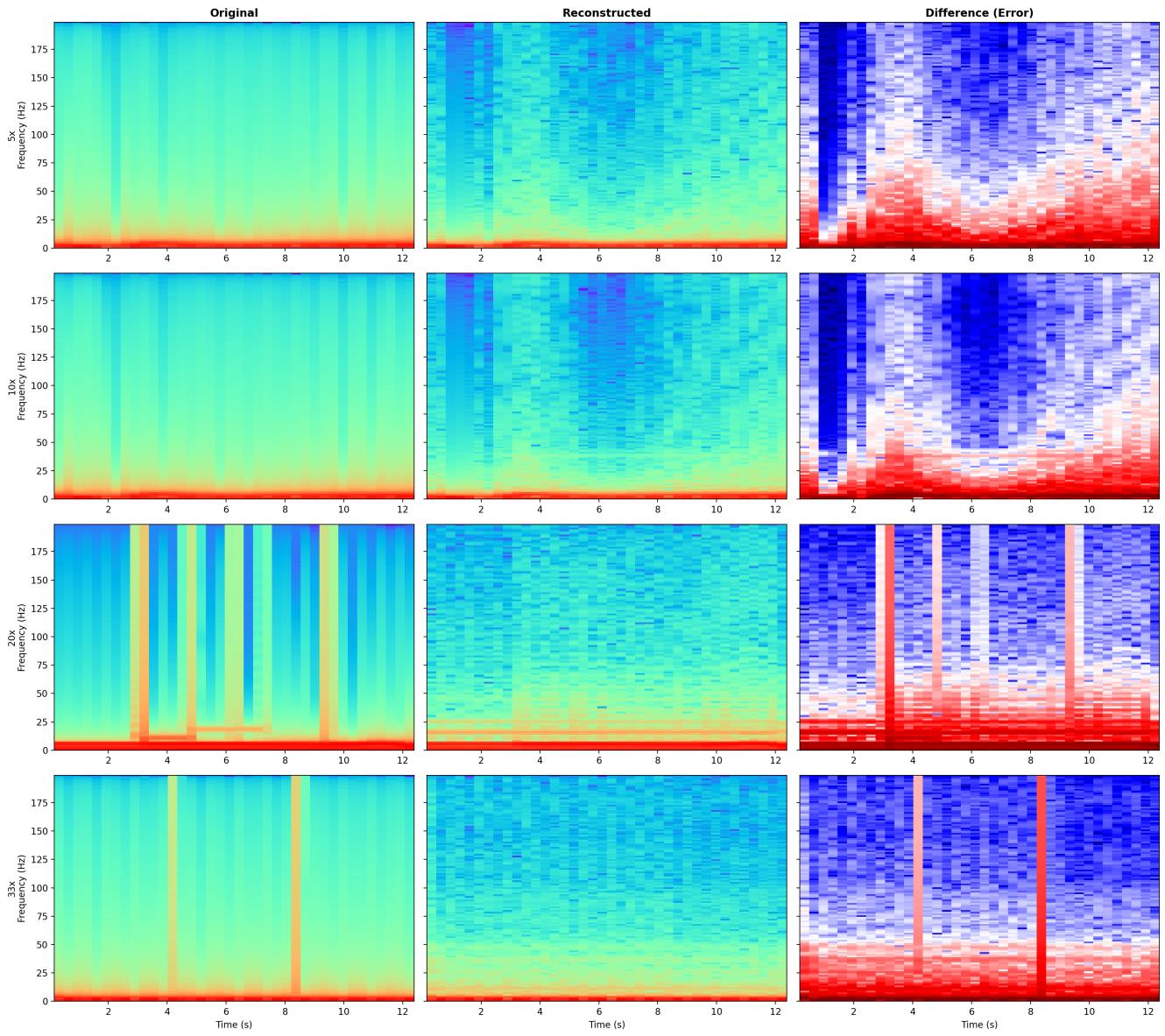
**Figure 12.** Training and validation loss evolution across all four upsampling factors ( $5\times$ ,  $10\times$ ,  $20\times$ ,  $33\times$ ). Each panel shows loss curves during training, demonstrating consistent convergence patterns and the absence of overfitting across all scaling factors. The systematic increase in final validation loss with upsampling factor reflects the inherent difficulty of reconstructing fine temporal details from severely undersampled inputs.

319 Figure ?? 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 ??) demonstrate  
 320 qualitative reconstruction fidelity, showing that models trained on CoSiBD can accurately recover high-frequency components,  
 321 temporal transitions, and amplitude dynamics even from extremely sparse inputs.

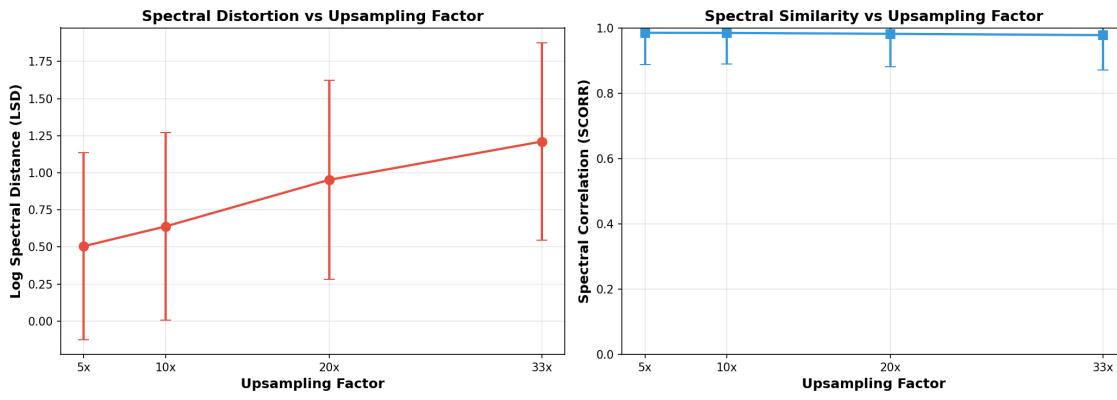


**Figure 13.** Representative prediction examples across all upsampling factors. Each quadrant shows prediction comparisons for a different scaling factor ( $5\times$ ,  $10\times$ ,  $20\times$ ,  $33\times$ ), displaying low-resolution inputs, ground-truth high-resolution signals, and CNN-reconstructed 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  $5\times$  upsampling to extreme  $33\times$  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 14.** Spectrogram comparison across all upsampling factors. Each row represents a different upsampling factor ( $5\times$ ,  $10\times$ ,  $20\times$ ,  $33\times$ ), 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.<sup>REV</sup>



**Figure 15.** Spectral quality metrics vs upsampling factor. Left: Log Spectral Distance (LSD) increases systematically with upsampling factor, from 0.51 ( $5\times$ ) to 1.21 ( $33\times$ ), 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.<sup>REV</sup>

### 327 Preliminary Application Results<sup>REV</sup>

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

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

337 Results demonstrate that synthetic data augmentation significantly improves model performance on real-world signals<sup>?</sup>.  
 338 For EEG validation, the Mixed strategy achieved MAE of  $9.73 \times 10^{-2}$ , representing a 9.64% improvement over the Real-only  
 339 baseline ( $10.77 \times 10^{-2}$ ). For out-of-domain VCTK speech data, the Tuned approach achieved MAE of  $4.41 \times 10^{-3}$ , a sub-  
 340 stantial 25.51% improvement over Real-only ( $5.92 \times 10^{-3}$ ). Notably, models trained exclusively on synthetic data (Synth-only)  
 341 exhibited higher errors, confirming that synthetic signals complement rather than replace real data<sup>?</sup>. These findings provide  
 342 quantitative evidence that CoSiBD successfully bridges the gap between synthetic training and real-world application, validating  
 343 its design objectives. Detailed experimental methodology, complete results, and model comparisons are documented in a  
 344 separate manuscript currently under preparation.<sup>REV</sup>

[REV 9]

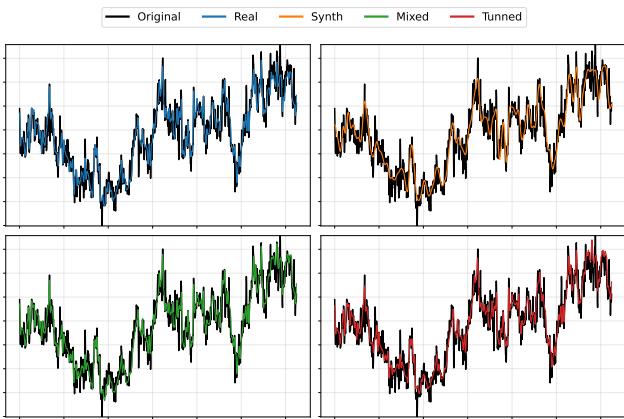
New subsection addressing review requirements for real-world application validation, CNN/deep learning model demonstration, and quantitative performance metrics.

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)	<b>9.73</b>	5.59
Tuned (pretrain + finetune)	10.68	<b>4.41</b>

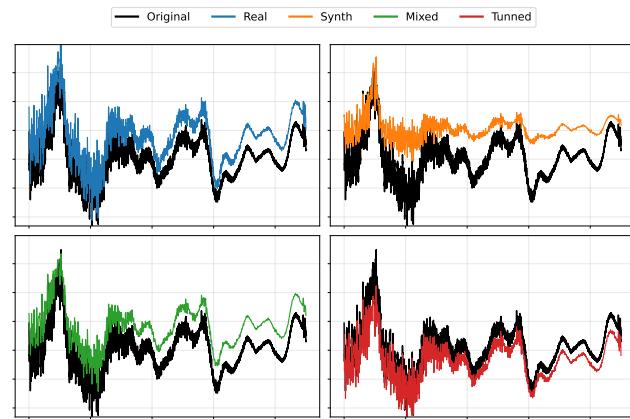
**Table 5.** Mean Absolute Error (MAE) for CNN-based super-resolution models trained with different strategies. Bold values indicate best performance for each dataset. Mixed strategy shows 9.64% improvement on EEG data, while Tuned strategy achieves 25.51% improvement on VCTK speech data, demonstrating the value of synthetic data augmentation.<sup>REV</sup>

### 347 Usage Notes

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



(a) EEG clinical signal reconstruction comparison.



(b) VCTK speech signal reconstruction comparison.

**Figure 16.** 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, Tuned). The comparisons demonstrate that synthetic data augmentation (Mixed and Tuned) produces reconstructions that more closely match the ground truth across both domains.<sup>REV</sup>

### 350    **Reading the Data**

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

```
354 import temana as tm
355
356 # Load low-resolution and high-resolution validation signals
357 x_valid = tm.read_data('SamplesAV_FV2024_07_09/SignalAVFV_Sub_Sample250_5000.txt')
358 y_valid = tm.read_data('SamplesAV_FV2024_07_09/SignalAVFV_Super_Sample1000_5000Val.txt')
```

359    These functions return PyTorch tensors representing the signals.

### 360    **Visualizing Signal Pairs**

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

```
362 import matplotlib.pyplot as plt
363
364 # Visualize the first pair of signals
365 plt.figure(figsize=(10, 4))
366 plt.plot(x_valid[0], label='Low-resolution (1000 points)', color='red')
367 plt.plot(y_valid[0], label='High-resolution (5000 points)', color='blue', alpha=0.7)
368 plt.xlabel('Time step')
369 plt.ylabel('Amplitude')
370 plt.title('Sample Signal Pair')
371 plt.legend()
372 plt.grid(True)
373 plt.tight_layout()
374 plt.show()
```

### 375    **Code availability**

376    Custom Python scripts used to load, process, and visualize the CoSiBD dataset are available at.<sup>REV</sup> The complete signal  
 377    generation pipeline, including modules for frequency profile generation, amplitude envelope construction, spline interpolation,

378 noise application, anti-aliasing filtering, and data export in multiple formats, is available at: <sup>REV</sup> CoSiBD scripts on GitHub. The  
379 repository includes SignalBuilderC, a modular Python package with documented functions for: (1) generating high-resolution  
380 signals with configurable parameters, (2) creating subsampled versions via re-evaluation or anti-aliasing filtering, (3) exporting  
381 signals in NumPy, text, and JSON formats, and (4) comprehensive metadata generation. All code is provided with example  
382 notebooks demonstrating dataset regeneration and usage.<sup>REV</sup> These scripts are distributed under the MIT License.

383  
384 The custom scripts are open access and provided under the Creative Commons Attribution 4.0 International (CC BY 4.0)  
385 license. The dataset itself is published separately at: [CoSiBD dataset on Zenodo](#).

[REV 10]  
Addresses reviewer requirement for comprehensive documentation and improved repository scriptation.

## 386 References

- 387 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](https://doi.org/10.1016/j.jelekin.2024.123456) (2024).
- 388 2. Nayak, S. K. *et al.* A review of methods and applications for a heart rate variability analysis. *Algorithms* **16**, 433, [10.3390/a16090433](https://doi.org/10.3390/a16090433) (2023).
- 389 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](https://doi.org/10.3389/fpubh.2017.00258) (2017).
- 390 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](https://doi.org/10.1109/VLSI-DAT54769.2022.9768053) (2022).
- 391 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](https://doi.org/10.48550/arXiv.1611.03530) (2016).
- 392 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](https://doi.org/10.1038/s42256-021-00321-8) (2021).
- 393 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](https://doi.org/10.1109/34.192463) (1989).
- 394 8. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444, [10.1038/nature14539](https://doi.org/10.1038/nature14539) (2015).
- 395 9. Goodfellow, I. J. *et al.* Generative adversarial networks. *arXiv preprint arXiv:1406.2661* [10.48550/arXiv.1406.2661](https://doi.org/10.48550/arXiv.1406.2661) (2014).
- 396 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).
- 397 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](https://doi.org/10.1016/j.dcn.2022.101067) (2022).
- 398 12. Schumaker, L. L. *Spline Functions: Basic Theory* (Springer-Verlag, New York, 2007), 3rd edn.
- 399 13. Boor, C. D. *A Practical Guide to Splines* (Springer-Verlag, New York, 2001).
- 400 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](https://doi.org/10.1145/3559540) (2023).
- 401 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).
- 402 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](https://doi.org/10.1029/2023MS003658) (2023).
- 403 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](https://doi.org/10.1038/s41592-023-02138-w) (2024).
- 404 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](https://doi.org/10.1038/s41587-025-02553-8) (2025).
- 405 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](https://doi.org/10.1109/TAU.1967.1161901) (1967).
- 406 20. Rabiner, L. R. & Gold, B. *Theory and Application of Digital Signal Processing* (Prentice Hall, 1975).

- 425 21. Marple, S. L., Jr. *Digital Spectral Analysis with Applications* (Prentice Hall, 1987).
- 426 22. Kuleshov, V., Enam, S. Z. & Ermon, S. Audio super resolution using neural networks. *arXiv preprint arXiv:1708.00853*  
427 [10.48550/arXiv.1708.00853](https://arxiv.org/abs/1708.00853) (2017).
- 428 23. Kaniraja, C. P., M, V. D. & Mishra, D. A deep learning framework for electrocardiogram (ecg) super resolution and  
429 arrhythmia classification. *Res. on Biomed. Eng.* **40**, 199–211, [10.1007/s42600-023-00320-x](https://doi.org/10.1007/s42600-023-00320-x) (2024).
- 430 24. Luciw, M. D., Jarocka, E. & Edin, B. B. Multi-channel eeg recordings during 3,936 grasp and lift trials with varying  
431 weight and friction. *Sci. Data* **1**, 140047, [10.1038/sdata.2014.47](https://doi.org/10.1038/sdata.2014.47) (2014).
- 432 25. Yamagishi, J., Veaux, C. & MacDonald, K. CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning  
433 toolkit (version 0.92), [10.7488/ds/2645](https://doi.org/10.7488/ds/2645) (2019).
- 434 26. Forestier, G., Petitjean, F., Dau, H. A., Webb, G. I. & Keogh, E. Generating synthetic time series to augment sparse datasets.  
435 In *2017 IEEE International Conference on Data Mining (ICDM)*, 865–870, [10.1109/ICDM.2017.106](https://doi.org/10.1109/ICDM.2017.106) (IEEE, 2017).

## 436 Acknowledgments

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

## 439 Author Contributions

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

## 443 Competing Interests

444 The authors declare no competing interests