Multilevel Graph Wavelet Compressed Sensing with Scale-Aware Neural Recovery

Amirhossein Nouranizadeh[+*] Sarang Patil[+†] Alan John Varghese[‡]
Varsha Narayanan[§] Amit Chakraborty[¶] Mengjia Xu[‖]

Abstract. Scientific machine learning methods such as neural operators and physics-informed neural networks have advanced engineering applications and inverse problems, but their training typically requires large volumes of simulated data. This makes data preparation and model training expensive. We propose Graph Wavelet Compressed Sensing (GWCS), a learning-based framework for offline compression of graph signals by representing them as sparse, interpretable wavelet-domain representations using the spectral graph wavelet transform. The framework combines a nonparametric multilevel importance sampler, which retains high-energy wavelet coefficients within each scale for a given compression ratio, with a scale-aware graph neural network that reconstructs the signal from the sparse coefficients. We evaluate the proposed framework on synthetic approximately band-limited graph signals over random graphs and four PDE simulation datasets over meshes, which include Turbulent Radiative Layer, Viscoelastic Instability, Kolmogorov Flow, and Dynamic Stall. We compare against graph signal sampling methods and graph autoencoder baselines. Results demonstrate that the framework achieves high reconstruction fidelity and substantial data compression compared to existing benchmarks.

1 Introduction Scientific machine learning (SciML) has emerged as a powerful paradigm for modeling complex physical, biological, and engineering systems. By integrating physics-based principles with data-driven inference, methods such as neural operators [24, 22] and physics-informed neural networks (PINNs) [31] have enabled the efficient approximation of partial differential equations (PDEs) and the solution of inverse problems across diverse scientific domains. However, these approaches typically rely on large volumes of high-fidelity simulation data for training, particularly when learning operators on fine spatial and temporal resolutions. This leads to substantial computational and storage costs, making scalability a major challenge for high-dimensional spatiotemporal systems. To address this, there has been a growing interest in learning efficient low-dimensional latent representations that can capture the essential structure of the underlying dynamics. This becomes even more challenging in settings involving unstructured data, such as meshes or irregular domains, which are naturally represented as graphs.

Compressed sensing (CS), a well-established mathematical theory widely used in domains such as medical image acquisition, exploits the underlying structure of natural signals to reconstruct an entire image from a few transform-domain samples by solving an optimization problem [9, 15]. In the graph signal processing domain, analogous graph signal sampling (GSS) techniques reconstruct graph signals from a few node- or spectral-domain samples by solving an optimization problem, typically assuming prior knowledge of the spectral properties of the signal under study [29]. These techniques motivate the design of a compression scheme that exploits the transform-domain sparsity already known to hold for physical fields on meshes and graphs [36, 32], without requiring per-signal optimization or prior knowledge of the signal's bandwidth.

In this work, inspired by CS and GSS theories, we propose Graph Wavelet Compressed Sensing (GWCS), a learning-based, offline graph signal compression technique that operates in the wavelet domain of graph-structured data. GWCS exploits the multiscale sparsity of graph wavelet coefficients to compress a signal through multilevel importance sampling of the coefficients, and trains a graph neural network (GNN) to reconstruct the full signal from the resulting sparse coefficients.

The main contributions are as follows.

(i) We propose **Graph Wavelet Compressed Sens-**

[*]Simplicial Technologies Inc., Toronto, ON, Canada (amirhossein@simplicial.ai).

[†]Department of Data Science, New Jersey Institute of Technology, Newark, NJ (sp3463@njit.edu).

[‡]School of Engineering, Brown University, Providence, RI (alan_john_varghese@brown.edu).

[§]Department of Computer Science, New Jersey Institute of Technology, Newark, NJ (vn299@njit.edu).

[¶]Siemens Foundational Technology, Princeton, NJ (amit.chakraborty@siemens.com).

[‖]Department of Data Science, New Jersey Institute of Technology, Newark, NJ (corresponding author: mx6@njit.edu).

[+] A. Nouranizadeh and S. Patil contributed equally to this work.

Code available at https://github.com/amrhssn/gwcs/.

ing (GWCS), a framework for learning-based compressed sensing of graph signals that combines a nonparametric **Multilevel Importance Sampling (MLIS)** module, which retains high-energy wavelet coefficients across scales, with a neural recovery model.

(ii) Within GWCS, we introduce the **Neural Inverse Graph Wavelet Transform (NIGWT)**, a scale-aware Encode-Process-Decode GNN that uses learnable scale embeddings and a residual IGWT connection to reconstruct signals from the sparse coefficients retained by MLIS.

(iii) To isolate the contribution of the wavelet domain, we also develop the **Sparse Graph Autoencoder (SGAE)**, a wavelet-free method with a learned sparse bottleneck, which we use throughout the experiments as a strong baseline for comparison against GWCS.

2 Related Works

2.1 Compressed Sensing. Compressed sensing (CS) has emerged as a fundamental framework for recovering signals from sub-Nyquist measurements by exploiting low-dimensional structure such as sparsity [15, 9]. Under suitable conditions, namely sparsity in a transform domain and incoherence between sampling and representation bases, accurate reconstruction can be achieved from a number of measurements proportional to the intrinsic sparsity of the signal, typically via ℓ_1-regularized optimization. These ideas have enabled a wide range of applications, including MRI, computerized tomography, and electron microscopy [25, 10, 23]. Real-world signals are only approximately sparse, incoherence conditions are rarely satisfied, and optimal sampling strategies are generally structured or variable-density rather than uniform random. These observations have motivated the community to extend the traditional CS theory into a new CS theory based on generalized assumptions of asymptotic sparsity, asymptotic incoherence, and multilevel random sampling [1, 2]. These works highlight the importance of structured representations, particularly wavelets, and adaptive sampling strategies.

Our goal is to compress graph signals, drawing inspiration from the multilevel sampling framework of CS theory. Unlike typical CS settings, where sampling occurs online at acquisition time and reconstruction is performed by solving a per-sample ℓ_1-regularized optimization problem, GWCS assumes access to the entire signal and its transform coefficients offline, and uses multilevel importance sampling to retain coefficients at multiple scales of the graph wavelet transform, and reconstructs the signal with a neural architecture. This enables compression for storage, transmission, and downstream processing.

2.2 Graph Signal Sampling. Graph signal sampling (GSS) is a family of graph signal processing techniques developed for sampling and reconstructing signals defined on irregular domains, where the goal is to select a subset of nodes that enable proper reconstruction of the full signal. Most approaches assume that graph signals are smooth or approximately band-limited with respect to a graph operator, and design sampling strategies in either the spectral or vertex domain [13, 3, 26]. Representative methods include greedy selection schemes, spectral proxy-based approaches that avoid eigendecomposition [3], and randomized strategies such as determinantal point processes [38], often accompanied by theoretical guarantees on recovery error. Despite their effectiveness, these methods exhibit several limitations. First, performance is highly sensitive to hyperparameters such as the assumed signal bandwidth or cutoff frequency, which are rarely known *a priori*. Second, they rely on fixed signal models (e.g., smoothness or band-limitedness), limiting their ability to adapt to complex, domain-specific structures. Finally, many approaches require per-signal optimization or expensive preprocessing, which hinders scalability to large datasets.

2.3 Graph Neural Networks. Graph neural networks (GNNs) provide a flexible framework for learning representations of graph-structured data through message passing [21, 18]. In the context of graph signals, GNNs have been used for reconstruction, denoising, and spatiotemporal prediction by learning data-driven priors that go beyond classical smoothness assumptions. Related directions include graph autoencoders for learning compact latent node representations [20], unrolled optimization networks for interpretable reconstruction [11], and multi-scale architectures that capture hierarchical dependencies [40, 17, 27]. In many SciML applications, the underlying data naturally resides on unstructured meshes that can be represented as graphs, making graph-based compression and reconstruction techniques particularly relevant. Our work builds on these ideas by combining multiscale graph representations with learning-based compressed sensing to enable efficient storage and reconstruction of scientific signals.

3 Preliminaries

3.1 Graph Spectral Foundations. We consider a data set of M samples, each consisting of an undirected graph and an associated signal, i.e., $\mathcal{D} = \{(\mathcal{G}^{(1)}, \mathbf{x}^{(1)}), \ldots, (\mathcal{G}^{(M)}, \mathbf{x}^{(M)})\}$. Each graph is defined as $\mathcal{G}^{(j)} = (\mathcal{V}^{(j)}, \mathcal{E}^{(j)})$, where $\mathcal{V}^{(j)}$ and $\mathcal{E}^{(j)}$ denote the node and edge sets; $N^{(j)} = |\mathcal{V}^{(j)}|$ is the number of nodes. Each graph is associated with a graph signal $\mathbf{x}^{(j)} \in \mathbb{R}^{N^{(j)}}$

defined over its nodes, where each entry corresponds to the feature value at a node. To simplify notation, we omit the sample index j and make it explicit when needed.

The symmetric normalized graph Laplacian matrix is defined as $\mathbf{L}_{\text{norm}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} \in \mathbb{R}^{N\times N}$, where $\mathbf{I}$ is the identity matrix, $\mathbf{A}$ is the adjacency matrix, and $\mathbf{D}$ is the diagonal degree matrix. The symmetric normalized Laplacian matrix has spectral decomposition: $\mathbf{L}_{\text{norm}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{U} = [\mathbf{u}_1 \dots \mathbf{u}_N] \in \mathbb{R}^{N\times N}$ is the orthonormal matrix of eigenvectors and $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_N) \in \mathbb{R}^{N\times N}$ is the diagonal matrix of eigenvalues and $0 = \lambda_1 \le \dots \le \lambda_N \le 2$. The eigenvectors form the graph Fourier basis, and the eigenvalues represent frequencies: small values correspond to smooth (low-frequency) signals over the graph, while large values correspond to rapidly varying (high-frequency) signals. The graph Fourier transform projects a graph signal $\mathbf{x}$ onto the spectral basis as $\tilde{\mathbf{x}} = \mathbf{U}^\top\mathbf{x}$, while the inverse reconstructs the signal via $\mathbf{x} = \mathbf{U}\tilde{\mathbf{x}}$ [37]. Graph spectral filtering is performed by applying a spectral filter $g : \mathbb{R} \to \mathbb{R}$ to the transformed signal, yielding $\tilde{\mathbf{y}} = g(\mathbf{\Lambda})\tilde{\mathbf{x}}$, where $g(\mathbf{\Lambda}) = \text{diag}(g(\lambda_1), \dots, g(\lambda_N))$. The filtered signal is then mapped back to the vertex domain as $\mathbf{y} = \mathbf{U}\tilde{\mathbf{y}} = \mathbf{U}g(\mathbf{\Lambda})\mathbf{U}^\top\mathbf{x}$.

3.2 Spectral Graph Wavelet Transform (SGWT). As a generalization of classical wavelet analysis to signals supported on irregular, non-Euclidean domains, the SGWT replaces geometric translation and scaling with filtering operations in the spectral domain of a graph operator such as the normalized Laplacian [19]. This enables a localized, multiscale analysis of graph signals without relying on an underlying Euclidean structure. In particular, SGWT constructs a spectral filterbank that decomposes signals into low-frequency components capturing global structure and high-frequency components capturing localized variations. To this end, we define a low-pass kernel $h : \mathbb{R} \to \mathbb{R}$ and a base band-pass kernel $g : \mathbb{R} \to \mathbb{R}$. From g we construct J band-pass kernels $g_s : \mathbb{R} \to \mathbb{R}$, $g_s(\lambda) := g(s\lambda)$, each parameterized by a scale value $s \in \{s_1, \dots, s_J\}$, where J is the number of band-pass scales. Each kernel satisfies $g_s(0) = 0$, $\lim_{\lambda\to\infty} g_s(\lambda) = 0$, and meets the admissibility condition of Lemma 5.1 in [19]. Given the spectral decomposition of the normalized Laplacian, the graph wavelet filterbank consists of a low-pass filter $\mathbf{\Psi}_0 = \mathbf{U}h(\mathbf{\Lambda})\mathbf{U}^\top$, and J band-pass filters $\mathbf{\Psi}_{s_1}, \dots, \mathbf{\Psi}_{s_J}$ where $\mathbf{\Psi}_{s_n} = \mathbf{U}g_{s_n}(\mathbf{\Lambda})\mathbf{U}^\top$. Stacking these graph filters yields the forward SGWT operator $\mathbf{\Psi} = \left[\mathbf{\Psi}_0^\top, \mathbf{\Psi}_{s_1}^\top, \dots, \mathbf{\Psi}_{s_J}^\top\right]^\top \in \mathbb{R}^{FN\times N}$, where $F = 1 + J$ is the number of filters in the wavelet filterbank. The graph wavelet coefficients are then obtained as $\mathbf{w} = \mathbf{\Psi}\mathbf{x} \in \mathbb{R}^{FN}$, which can be reshaped as a 2D matrix $\mathbf{W} \in \mathbb{R}^{N\times F}$, where each node has F graph wavelet coefficients across different frequency bands. The graph signal can be recovered by the inverse SGWT, i.e., $\mathbf{x} = \mathbf{\Psi}^\dagger\mathbf{w}$, where $\mathbf{\Psi}^\dagger := (\mathbf{\Psi}^\top\mathbf{\Psi})^{-1}\mathbf{\Psi}^\top$ is the pseudo-inverse of $\mathbf{\Psi}$.

3.3 Chebyshev polynomial approximation of graph wavelet coefficients. Spectral filtering in SGWT requires the full eigendecomposition of the normalized graph Laplacian with cubic computational cost, which makes it impractical for large graphs. In practice, we use truncated Chebyshev polynomial series to approximate spectral filters $h(\lambda)$ and $g_{s_j}(\lambda)$ as polynomial filters of normalized Laplacian that can be applied directly to the graph signal in the node domain efficiently [19], see the Appendix A for computational details.

4 Method In this section, we provide a detailed description of the proposed Graph Wavelet Compressed Sensing (GWCS) framework and Sparse Graph Autoencoder (SGAE).

GWCS and SGAE are both controlled by a single **target compression ratio** $c \in (0, 1]$, which is the fraction of wavelet coefficients (GWCS) or latent entries (SGAE) kept relative to the size of the input signal. For a graph signal with N nodes and C channels, this gives a retention budget $m = cNC$. We set $c = 5\%$ for all experiments conducted in this paper.

The GWCS framework in Figure 4.1 includes two main modules: (i) a nonparametric *Multilevel Importance Sampling (MLIS)* module that retains a fixed budget of wavelet coefficients using an energy-based heuristic informed by the multilevel structure of the wavelet transform, and, (ii) the *Neural Inverse Graph Wavelet Transform (NIGWT)* module, a scale-aware GNN recovery model that reconstructs the full signal from the sparse coefficients.

4.1 Multilevel Importance Sampling Module. Inspired by the multilevel random subsampling technique introduced in the compressed sensing theory [1, 33], we design our *Multilevel Importance Sampling (MLIS)* module in the graph wavelet domain, which is responsible for subsampling the graph wavelet coefficients from observed data.

Given the wavelet coefficient matrix $\mathbf{W} \in \mathbb{R}^{N\times F}$ and a sampling budget $m = cN$, MLIS exploits the multilevel energy structure to allocate the budget across scales and select the most informative coefficients within each scale. Throughout this section, $W_{i,k}$ denotes the (i, k) entry of $\mathbf{W} \in \mathbb{R}^{N\times F}$ (node i, scale k). The energy of scale $k \in \{1, \dots, F\}$ is

$$E_k = \sum_{i=1}^{N} W_{i,k}^2, \tag{4.1}$$

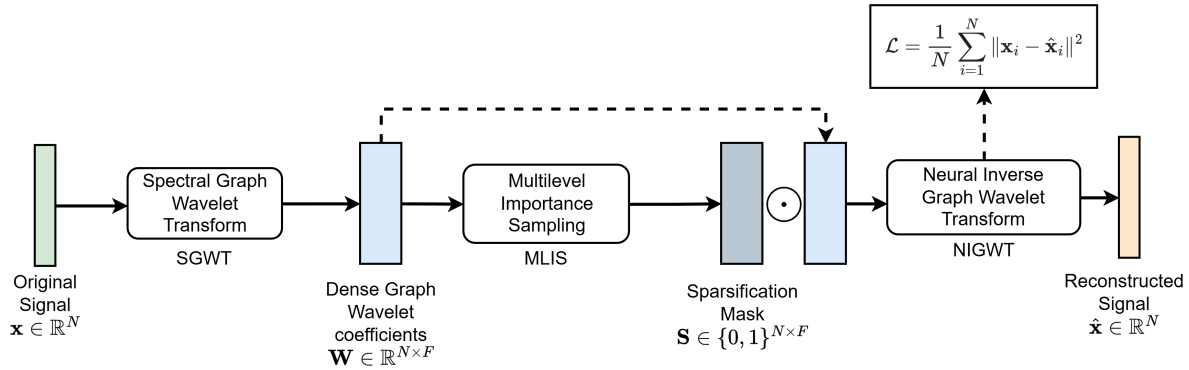


Figure 4.1: Overview of the GWCS Framework. 1) The input graph signal $\mathbf{x} \in \mathbb{R}^N$ is first transformed by the Spectral Graph Wavelet Transform (SGWT), implemented via a Chebyshev approximation of degree $K = 100$, to obtain a dense wavelet coefficient matrix $\mathbf{W} \in \mathbb{R}^{N\times F}$ ($F = 5$ corresponds to 1 lowpass and $J = 4$ band-pass filters). 2) The Multilevel Importance Sampling (MLIS) module allocates a retention budget of $m = cN$ coefficients across scales in proportion to each scale's energy (Eq. (4.2)), then retains coefficients within each scale via importance sampling with probability proportional to squared coefficient magnitude (Eq. (4.3)), yielding the binary sparsification mask $\mathbf{S} \in \{0,1\}^{N\times F}$ and the sparse coefficient matrix $\tilde{\mathbf{W}} \in \mathbb{R}^{N\times F}$. 3) The Neural Inverse Graph Wavelet Transform (NIGWT) module recovers the full signal $\hat{\mathbf{x}} \in \mathbb{R}^N$ via an "Encode-Process-Decode" architecture with a residual IGWT connection.

and the fraction of the total budget allocated to scale k is

$$r_k = \frac{E_k}{\sum_{k'=1}^{F} E_{k'}}, \qquad m_k = \lfloor r_k \cdot m \rfloor, \tag{4.2}$$

so that $\sum_{k=1}^{F} m_k = m$ and scales with higher energy receive more samples. For importance-weighted intra-scale sampling within scale k, we draw m_k node indices without replacement according to the probability,

$$p_{i,k} = \frac{W_{i,k}^2}{\sum_{i'=1}^{N} W_{i',k}^2}, \tag{4.3}$$

concentrating measurements on nodes with larger-magnitude coefficients at that scale. Intuitively, dropping a large-magnitude coefficient increases the reconstruction error more than dropping a near-zero one, so retaining high-energy nodes is a natural proxy for minimizing reconstruction error under a fixed budget. The procedure yields a binary mask $\mathbf{S} \in \{0,1\}^{N\times F}$, where $S_{i,k} = 1$ if coefficient (i,k) is selected, with $\|\mathbf{S}\|_0 = m$. The sparse coefficient matrix is then obtained as

$$\tilde{\mathbf{W}} = \mathbf{S} \odot \mathbf{W} \in \mathbb{R}^{N\times F}, \tag{4.4}$$

where $\odot$ is the Hadamard product. The pair $(\tilde{\mathbf{W}}, \mathbf{S})$ is the compressed representation fed to the reconstruction model.

4.2 Neural Inverse Graph Wavelet Transform (NIGWT). NIGWT reconstructs $\hat{\mathbf{x}} \in \mathbb{R}^N$ from $(\tilde{\mathbf{W}}, \mathbf{S}, \mathcal{G})$ using an Encode-Process-Decode architecture [35, 8], implementing learned scale embeddings and a residual IGWT connection. Let D denote the hidden dimension and σ the SiLU activation.

Encoder. The Encoder network computes an initial node embedding matrix by combining sparse graph wavelet coefficients $\tilde{\mathbf{W}}$, sparsification mask $\mathbf{S}$, and learnable scale encodings $\mathbf{E}$. Specifically, we first compute the encoded inputs by transforming the concatenation of the sparse graph wavelet coefficients and the selection mask, $\tilde{\mathbf{W}} \| \mathbf{S} \in \mathbb{R}^{N\times 2F}$:

$$\mathbf{Z}^{\text{input}} = \sigma\Big(f_{\text{Enc}}^{(1)}\Big(\tilde{\mathbf{W}} \| \mathbf{S}\Big)\Big), \tag{4.5}$$

where $f_{\text{Enc}}^{(1)} : \mathbb{R}^{2F} \to \mathbb{R}^D$ is an MLP. The main element of the Encoder is the learned scale-encodings. Similar to positional encodings in the Transformer architecture [39], which provide tokens with an explicit notion of position, our learned scale-encodings give each graph wavelet scale an explicit, learnable identity that the network can use to distinguish coarse and fine scales. Specifically, we learn a set of scale-encodings $\mathbf{e}_1, \ldots, \mathbf{e}_F \in \mathbb{R}^D$ that act as basis vectors for scale information; stacking them gives $\mathbf{E} \in \mathbb{R}^{F\times D}$. For each node i we compute the node-scale embedding $\mathbf{z}_i^{\text{scale}} \in \mathbb{R}^D$ by a weighted sum over scales:

$$\mathbf{z}_i^{\text{scale}} = \sum_{k=1}^{F} \tilde{W}_{i,k}\, \mathbf{e}_k, \quad \text{i.e.,} \quad \mathbf{Z}^{\text{scale}} = \tilde{\mathbf{W}}\mathbf{E} \in \mathbb{R}^{N\times D}, \tag{4.6}$$

using the retained coefficient magnitude $\tilde{W}_{i,k}$ as the weight, so that $\mathbf{z}_i^{\text{scale}}$ encodes how much energy node i

carries at each scale. Finally, we concatenate the node input and scale embedding matrices and apply the last transformation to get the initial node embeddings:

$$\mathbf{Z}^{(0)} = \sigma\Big(f_{\mathsf{Enc}}^{(2)}\big(\mathbf{Z}^{\text{input}} \| \mathbf{Z}^{\text{scale}}\big)\Big), \quad \mathbf{Z}^{(0)} \in \mathbb{R}^{N\times D}, \tag{4.7}$$

where $f_{\mathsf{Enc}}^{(2)} : \mathbb{R}^{2D} \to \mathbb{R}^{D}$ is an MLP.

Processor. The Processor network comprises L message-passing layers that process the initial node embedding matrix and produces informative node representations for signal reconstruction. In our implementation, each layer $f_{\mathsf{Proc}}^{(\ell)}$ is a GraphSAGE convolution (SAGEConv) [18] preceded by layer normalization $\mathrm{LN}(\cdot)$ and skip connection:

$$\mathbf{Z}^{(\ell+1)} = f_{\mathsf{Proc}}^{(\ell)}\Big(\sigma\big(\mathrm{LN}\big(\mathbf{Z}^{(\ell)}\big)\big), \mathcal{G}\Big) + \mathbf{Z}^{(\ell)}, \quad \ell = 0, \ldots, L-1. \tag{4.8}$$

Decoder. The final component transforms node representations to the reconstructed signal. This module learns the residual error between the original graph signal and its reconstruction obtained via the pseudo-inverse graph wavelet transform:

$$\hat{\mathbf{x}} = \mathrm{IGWT}(\tilde{\mathbf{W}}) + f_{\mathsf{Dec}}\big(\mathbf{Z}^{(L)}\big), \tag{4.9}$$

where $\mathrm{IGWT}(\tilde{\mathbf{W}}) = \mathbf{\Psi}^{\dagger}\mathrm{vec}(\tilde{\mathbf{W}}) \in \mathbb{R}^{N}$ is the Chebyshev-approximated inverse SGWT (computed without gradient updates), and $f_{\mathsf{Dec}} : \mathbb{R}^{D} \to \mathbb{R}$ is an MLP applied node-wise. The residual formulation allows NIGWT to refine an initial IGWT-based reconstruction, leading to improved optimization stability and faster convergence.

4.3 Sparse Graph Autoencoder (SGAE).

To isolate the contribution of the wavelet domain, we introduce the Sparse Graph Autoencoder (SGAE) as a baseline that operates directly on the raw signal without any SGWT preprocessing or multilevel sampling, shown in Figure 4.2.

Sparse Encoder (Mask Generation Model). A stack of L SAGEConv layers encodes the raw signal $\mathbf{x} \in \mathbb{R}^{N}$ into a dense latent $\mathbf{Z} \in \mathbb{R}^{N\times D_{\mathrm{SGAE}}}$. A differentiable sparse bottleneck then retains $m = cN$ latent entries via Straight-Through Estimation (STE) with temperature annealing:

$$\tilde{\mathbf{Z}} = \mathrm{SparseBottleneck}(\mathbf{Z}, \rho), \quad \|\tilde{\mathbf{Z}}\|_0 = cN, \tag{4.10}$$

where $c = 5\%$ is the same target compression ratio used for GWCS.

Graph Decoder (Reconstruction Model). A symmetric stack of SAGEConv layers maps $\tilde{\mathbf{Z}}$ back to the signal domain: $\hat{\mathbf{x}} = \mathrm{Decoder}(\tilde{\mathbf{Z}}, \mathcal{E})$. Crucially, SGAE learns a purely data-driven compressed representation as it bypasses the SGWT entirely and has no inductive bias towards wavelet sparsity.

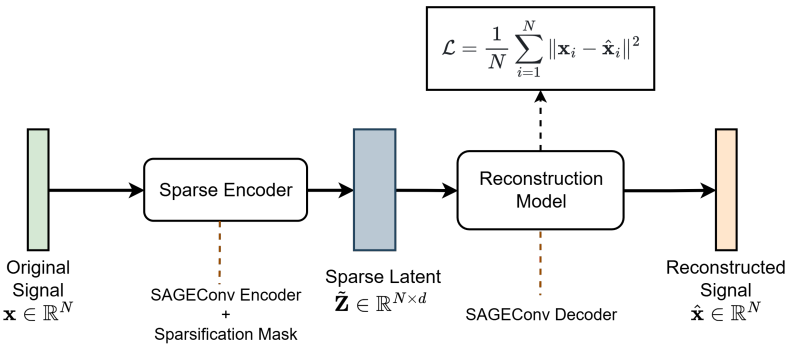


Figure 4.2: Overview of the SGAE Framework. The raw signal $\mathbf{x} \in \mathbb{R}^N$ is encoded directly by the Sparse Encoder into a sparse latent $\tilde{\mathbf{Z}} \in \mathbb{R}^{N\times D_{\mathrm{SGAE}}}$ ($D_{\mathrm{SGAE}} = 128$; cN entries active). The Reconstruction Model (Graph Decoder) maps $\tilde{\mathbf{Z}}$ back to the signal domain.

Training Objectives. Both GWCS and SGAE are trained end-to-end by minimizing the same MSE reconstruction loss over the dataset with the Stochastic Gradient Descent (SGD). For GWCS, let θ denote the parameters of the NIGWT module. The reconstructed signal is given by $\hat{\mathbf{x}}^{(j)}(\theta) := \mathrm{NIGWT}_{\theta}(\mathrm{MLIS}(\mathbf{W}), \mathcal{G})$, and the MSE objective is

$$\theta^{\star} = \arg\min_{\theta} \frac{1}{M}\sum_{j=1}^{M} \frac{1}{N^{(j)}} \big\|\mathbf{x}^{(j)} - \hat{\mathbf{x}}^{(j)}(\theta)\big\|_2^2. \tag{4.11}$$

SGAE is trained with the identical objective over its encoder-decoder parameters, with gradients passed through the sparse bottleneck via STE.

5 Experiments

5.1 Datasets. Synthetic Approximately Band-limited (ABL) signals. We employ a class of ABL graph signals with additive Gaussian noise, modeled as $\mathbf{x} = \mathbf{U}_R\mathbf{a} + \mathbf{n}$, where $\mathbf{U}_R \in \mathbb{R}^{N\times R}$ contains the R lowest-frequency eigenvectors of the normalized Laplacian $\mathbf{L}$, $\mathbf{a} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$ is a vector of band-limited coefficients, and $\mathbf{n}$ is additive Gaussian noise. Signals consistent with this model have most of their energy in the low-frequency components. We generate random graphs from the grid, Erdős–Rényi [16], and Barabási–Albert [7] models with $N \in [100, 625]$, and for each graph generate a random band-limited signal with additive noise. Details of the data-generating process are provided in Appendix B.

Physics-based PDE Datasets. We further evaluate our model on four PDE datasets drawn from the WELL benchmark [28] and the PBFM benchmark [6], both providing standardized, high-resolution PDE simulations for SciML.

Table 4.1: Reconstruction performance on the ABL dataset. Our method achieves superior performance while using significantly fewer samples (1-10% of wavelet coefficients with $F = 5$ filters and a single-channel signal, at $c = 5 - 50\%$ of nodes). All GSS baselines use oracle bandwidth knowledge and optimized hyperparameters. Best results per column in bold.

	Compression Ratio c	**Low** ($c = 5\%$) = 1% coeffs		**Medium** ($c = 25\%$) = 5% coeffs		**High** ($c = 50\%$) = 10% coeffs	
Method		**MSE(x)**	**MSE(clean)**	**MSE(x)**	**MSE(clean)**	**MSE(x)**	**MSE(clean)**
ESS	5% / 25% / 50% nodes	0.18±0.16	0.14±0.15	0.05±0.03	0.02±0.02	0.039±0.027	**0.007±0.007**
RSBS	5% / 25% / 50% nodes	0.25±0.19	0.21±0.18	0.13±0.11	0.10±0.10	0.078±0.066	0.064±0.060
FastGSSS	5% nodes only	0.19±0.16	0.15±0.15	—	—	—	—
BSGDA	5% / 25% / 50% nodes	0.21±0.17	0.18±0.16	0.22±0.18	0.20±0.17	0.128±0.112	0.132±0.111
SGAE	same # latent coeffs	0.194±0.141	0.163±0.126	0.105±0.069	0.105±0.058	0.027±0.020	0.049±0.025
GWCS (Ours)	1% / 5% / 10% coeffs	**0.13±0.10**	**0.09±0.09**	**0.04±0.02**	**0.019±0.019**	**0.021±0.014**	0.013±0.009

Table 4.2: Reconstruction RMSE (physical units, mean ± std) on four PDE datasets at compression ratio $c = 5\%$. **Bold**: best per dataset; underline: second-best. ‡Classical GSS values are computed over 50 test samples, as full evaluation is computationally prohibitive (≈5–6 h per sample). Supplementary MSE and relative ℓ_2 metrics are in Appendix D.

Method	**Viscoelastic**	**Turbulent**	**Kolmogorov**	**Dynamic Stall**
FastGSSS‡	0.0051 ± 0.0010	40.5 ± 6.9	0.834 ± 0.110	9310 ± 1908
BSGDA‡	0.0040 ± 0.0006	8.9 ± 1.5	0.508 ± 0.070	5368 ± 1040
RSBS‡	0.0044 ± 0.0007	9.2 ± 1.6	0.563 ± 0.082	6092 ± 1203
GXN	0.0052 ± 0.0040	41.5 ± 1.5	0.851 ± 0.113	9350 ± 1861
SGAE-50K	$\underline{0.0035 \pm 0.0026}$	$\underline{5.72 \pm 2.71}$	$\mathbf{0.287 \pm 0.059}$	$\underline{3211 \pm 819}$
SGAE-99K	0.0037 ± 0.0030	$\mathbf{4.78 \pm 2.26}$	0.328 ± 0.050	$\mathbf{2480 \pm 744}$
GWCS (Ours)	$\mathbf{0.0016 \pm 0.0011}$	7.06 ± 1.09	$\underline{0.316 \pm 0.041}$	9119 ± 1700

Turbulent Radiative Layer 2D ($N = 49{,}152$). This dataset simulates cold dense gas clumps moving through a surrounding hotter gas, mixing due to turbulence at their interface in astrophysical environments. The mixing creates an intermediate temperature phase that rapidly cools by radiative cooling, causing the mixed gas to join the cold phase. We compress and reconstruct the **density** field.

Viscoelastic Instability ($N = 91{,}136$). This dataset captures the evolution of viscoelastic fluid flows exhibiting elastic instabilities, governed by coupled nonlinear PDEs describing the interaction between velocity and stress fields, with complex multiscale spatiotemporal dynamics. We reconstruct the **pressure** field.

Kolmogorov Flow ($N = 16{,}384$). This dataset represents a canonical turbulent flow driven by a sinusoidal forcing, commonly used as a benchmark for studying transition to turbulence and chaotic dynamics, with rich vortex structures and multiscale behavior. We reconstruct the $\mathbf{velocity}_x$ field.

Dynamic Stall ($N = 16{,}384$). This dataset models unsteady aerodynamic flows around an airfoil undergoing rapid changes in angle of attack, leading to flow separation and vortex shedding. The data captures highly nonlinear transient phenomena relevant to aerospace applications. We reconstruct the **pressure** field.

5.2 Baselines. Graph signal sampling (GSS).

We compare against four classical non-parametric GSS algorithms that sequentially select optimal node subsets and provide theoretical reconstruction guarantees: ESS [4], RSBS [30], FastGSSS [34], and BSGDA [5], all implemented via the `THGSP` package [14]. To ensure a fair comparison, we conduct comprehensive hyperparameter optimization for all GSS methods on the validation split of the ABL dataset. Since GSS methods are non-parametric and operate on individual graph signals without training, we use grid search over method-specific hyperparameter spaces to minimize validation mean-squared error (MSE). Details of the hyperparameter search of GSS methods can be found in Appendix C. Additionally, we provide GSS methods with oracle knowledge of the true signal bandwidth R that represents an optimistic baseline.

GXN. A joint neural sampling-and-recovery baseline: NeuralSampling selects = 5% of nodes via mutual information-based node scoring; NeuralRecovery

https://github.com/bwdeng20/thgsp

reconstructs the full signal via a graph polynomial network [12].

SGAE. Our sparse graph autoencoder (Section 4.3), is evaluated at two parameter budgets: **SGAE-99K** (≈99,000 parameters, latent dimension $D_{\text{SGAE}} = 128$, $L = 4$ SAGEConv encoder/decoder layers) and **SGAE-50K** (≈50,000 parameters), both at target compression ratio $c = 5\%$ matched to GWCS.

GWCS. Our proposed model uses hidden dimension $D = 256$, $L = 5$ SAGEConv layers, $F = 5$ wavelet filters ($J = 4$ bandpass plus one lowpass), Chebyshev degree $K = 100$, yielding ≈993,000 parameters. A lightweight variant ($D = 96$, $L = 3$, ≈99,000 parameters) is used for parameter-matched comparison against SGAE-99K. All models use a compression ratio $c = 5\%$.

5.3 Evaluation. We report the Root Mean Squared Error (RMSE) in physical units, computed after de-normalizing predictions to the original signal scale. The sample-level RMSE for sample j is given by $\text{RMSE}^{(j)} = \|\hat{\mathbf{x}}^{(j)} - \mathbf{x}^{(j)}\|_2/\sqrt{N}$, and we compute the mean and standard deviation across all test samples.

5.4 Results on ABL Signals. Table 4.1 reports MSE across three compression levels. GWCS outperforms all GSS baselines (ESS, RSBS, FastGSSS, and BSGDA) at the low and medium budgets by a large margin *without* oracle bandwidth knowledge, and matches the best GSS method (ESS) at the high budget. SGAE also underperforms GWCS at all budgets on ABL signals, confirming that the wavelet-domain representation provides a strong inductive bias for approximately band-limited data. Figures E.1–E.4 in Appendix E further visualize the SGWT coefficients, MLIS sparsification mask, and NIGWT reconstruction for four representative ABL test samples over grid and Erdős–Rényi graphs.

5.5 Results on PDE Datasets. Reconstruction performance across the four PDE simulation datasets, measured by RMSE, is shown in Table 4.2. The central finding is that, GWCS excels on spectrally structured PDE fields and is competitive on moderate turbulence, while SGAE is preferable when energy is distributed broadly across wavelet scales. Supplementary quantitative metrics (MSE and relative ℓ_2 error) for all neural models are reported in Appendix D, confirming the same ordering across metrics. This pattern is entirely consistent with the theoretical motivation of GWCS: wavelet-domain sampling provides a decisive advantage precisely when the signal's energy is concentrated in a small number of scales. Figure 5.1 shows qualitative reconstruction of the Turbulent Radiative Layer density field on a representative test sample (sample 48) at compression ratio $c = 5\%$. Figure E.5(a) shows that GWCS recovers the dominant hot–cold density interface: the sharp boundary between the cold dense phase and the surrounding hot gas is well preserved, and the residual absolute error is concentrated along the turbulent mixing layer boundary where broadband fluctuations exceed the capacity of the fixed five-filter SGWT at $c = 5\%$. Figures 5.1(b) and (c) show that SGAE-99K and SGAE-50K achieve lower per-sample RMSE; their unconstrained, data-driven latent representations capture the broadband energy components that GWCS's fixed wavelet filterbank discards. The larger capacity SGAE-99K (≈99K parameters) outperforms SGAE-50K (≈50K) on this dataset, confirming that representational capacity matters for broadband flows. Figure 5.1(d) illustrates the failure of GXN: the reconstruction is nearly uniform, failing entirely to resolve the density contrast, consistent with its mean RMSE of 41.49 across the full test set. The root cause is that turbulent radiative layer flows have broad-band energy distributed across all wavelet scales; the strict 1% wavelet coefficient budget in GWCS cannot retain enough energy in all scales simultaneously, whereas SGAE's learned latent space is unconstrained and can allocate capacity freely across the signal's energy distribution. Qualitative reconstructions for all four datasets and baselines are provided in Appendix E.

Figure 5.2 shows per-sample RMSE trajectories for two representative cases. Panel (a) covers all test samples for Viscoelastic Instability using neural models only; classical GSS baselines are omitted because their per-sample cost is computationally expensive on this large mesh (N=91,136 nodes). Panel (b) shows the first 50 Kolmogorov test samples with both neural and classical baselines; the truncation to 50 samples avoids visual crowding with eight overlapping series and matches the subset for which classical methods were evaluated. For Viscoelastic Instability, GWCS achieves the lowest RMSE (0.0016), outperforming SGAE-50K (×2.2 better), SGAE-99K (×2.3), and GXN (×3.2). Viscoelastic flows have spectrally structured spatial patterns governed by smooth PDEs; the SGWT captures this structure efficiently and GWCS's scale-aware encoder and residual IGWT connection leverage it strongly. For Kolmogorov Flow, SGAE-50K achieves the lowest RMSE (0.287), with GWCS (0.319) and SGAE-99K (0.328) close behind; GXN substantially underperforms (0.851). Notably, the smaller SGAE-50K outperforms the larger SGAE-99K on this dataset, suggesting that the moderate-turbulence flow does not require the additional representational capacity and that the 99K variant may be slightly overfit. As shown in Figure 5.2(b), all three neural methods cluster well below the classical GSS baselines (FastGSSS, RSBS, BSGDA, RMSE ≈ 0.6–1.1), confirming the value of

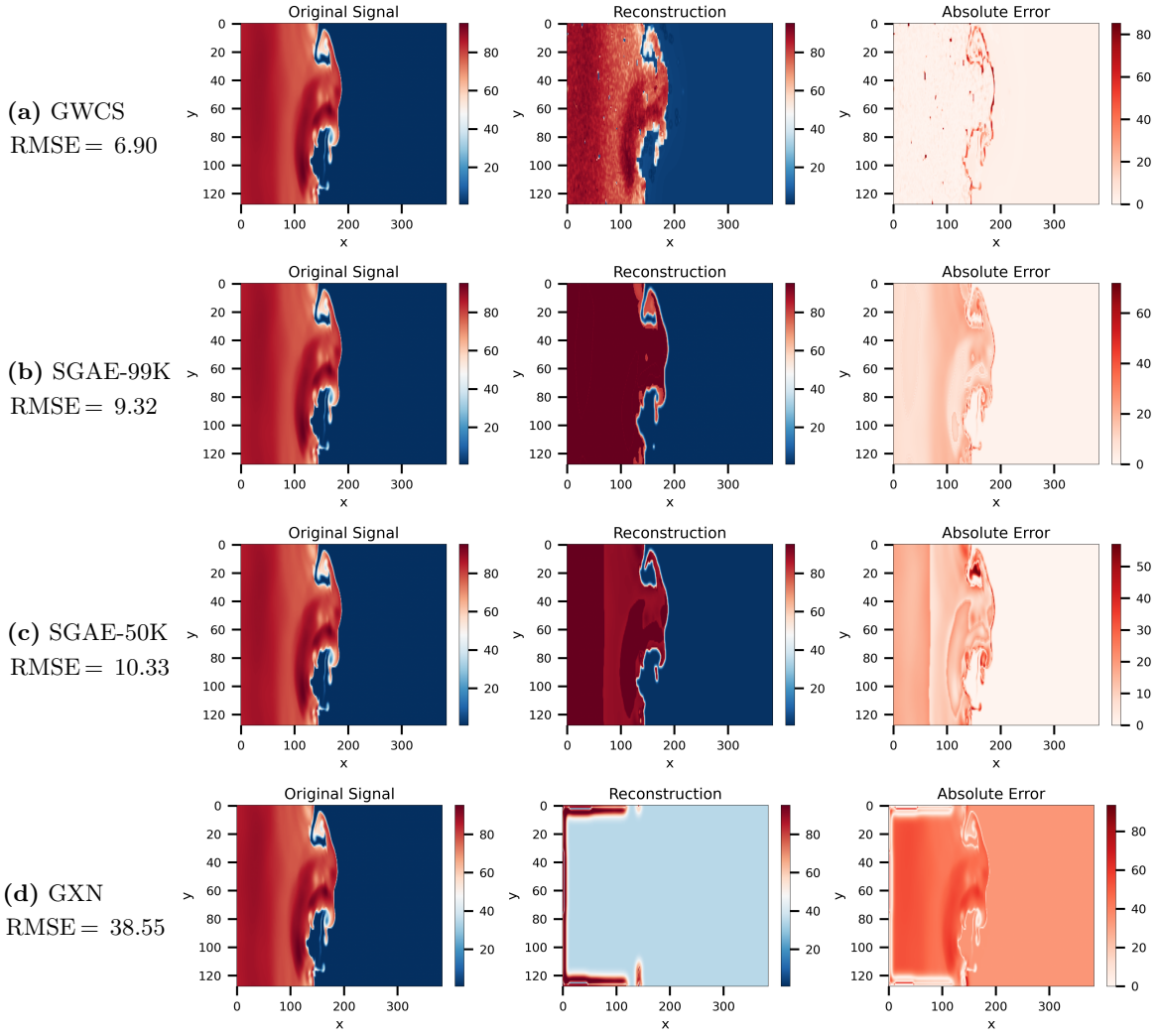
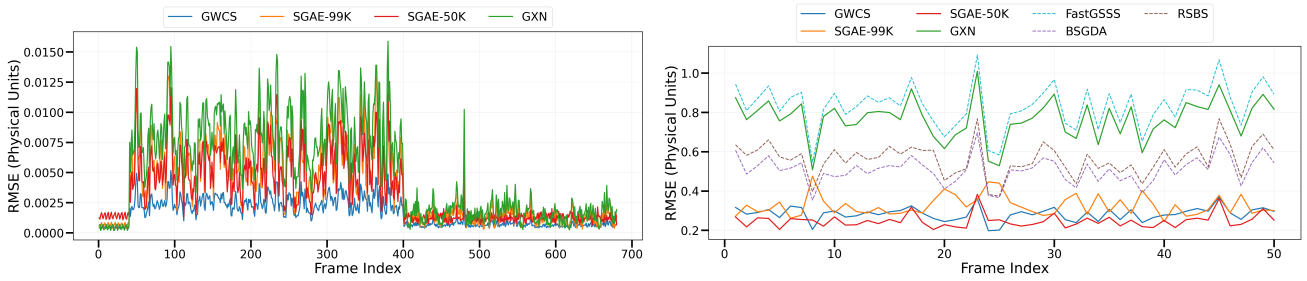


Figure 5.1: Qualitative reconstruction of the Turbulent Radiative Layer density field (test sample 48, compression ratio $c = 5\%$). Columns: ground truth, reconstruction, absolute error.



(a) Viscoelastic Instability: per-sample RMSE (all test samples; neural models only)

(b) Kolmogorov Flow: per-sample RMSE (first 50 test samples; neural + classical baselines)

Figure 5.2: Per-sample RMSE trajectories (physical units). **(a) Viscoelastic Instability** (all test samples, neural models only): GWCS maintains the lowest RMSE throughout the test set. **(b) Kolmogorov Flow** (first 50 samples, neural and classical): all three sparsity-based models cluster well below GXN and classical GSS baseline (FastGSSS, RSBS, BSGDA), confirming the necessity of learned representations.

learned representations for turbulent flows. GWCS is competitive with SGAE despite its structured wavelet prior operating at the same $c = 5\%$ compression ratio.

For Turbulent Radiative Layer, SGAE-99K achieves the best mean RMSE (4.78), with GWCS second (7.06), SGAE-50K third (5.72), and GXN failing entirely (41.5). Turbulent flows have broad-band energy across all wavelet scales; the strict 1% wavelet coefficient budget ($c = 5\%$) in NIGWT discards energy that SGAE's data-driven latent representation retains. For Dynamic Stall dataset, SGAE-99K dominates in terms of lowest reconstruction error; whereas GWCS and GXN perform substantially poorly. Dynamic stall features sharp transient spatial gradients at the airfoil leading edge; the fixed 5-filter SGWT may not resolve these structures within a 1% coefficient budget ($c = 5\%$). Across all four datasets, sparsity methods (GWCS and SGAE) consistently outperform classical GSS and GXN, validating the learning-based compressed sensing approach. GWCS has a decisive advantage on spectrally regular PDE data (Viscoelastic), is competitive on moderate turbulence, and is outperformed by SGAE on highly complex or transient datasets. This pattern is consistent with the theoretical motivation: wavelet-domain sampling benefits signals whose energy is concentrated in a small number of scales and spatial locations.

5.6 Discussion. GWCS supports both *transductive* and *inductive* modes of use. In the transductive setting, GWCS compresses an entire dataset for storage or transmission at a given compression ratio, producing a sparse wavelet-domain representation of the dataset along with a trained model that can later reconstruct the full signals. In the inductive setting, GWCS is trained once and then applied to unseen data as a sampler and reconstructor for downstream machine learning. MLIS compresses new test signals into sparse graph wavelet coefficients that downstream processing can consume directly, and NIGWT recovers the original domain signal only when it is needed.

GWCS differs from classical CS along two axes. Classical CS is posed *online*, with sampling designed at acquisition time, whereas GWCS is *offline*, operating on signals that are already fully available. Classical CS also reconstructs each sample by solving an ℓ_1-regularized optimization problem with no learning capacity, whereas GWCS trains a neural model to reconstruct signals from their retained coefficients. The online setting in CS applications is motivated by the cost of physical measurements, a cost that is largely irrelevant in SciML, where data typically consists of large volumes of simulated PDE solutions already available offline. This offline setting instead incurs the cost of storing the full signal and its coefficients, and the computational cost of the Chebyshev-approximated SGWT. In exchange, once trained, reconstruction requires only a single MLIS pass and a forward pass of NIGWT.

MLIS is a heuristic sampler that exploits the sparsity of natural signals in the wavelet domain together with multilevel, energy-based coefficient retention, rather than learning which coefficients to keep. Compared to learning-based samplers such as GXN and SGAE, this yields an interpretable compressed representation and a more stable training process for the reconstruction model. This heuristic performs best when a signal's energy is concentrated in a few wavelet scales, as for Viscoelastic Instability, where GWCS achieves its best reconstruction among the four PDE datasets. When energy is instead distributed broadly across scales, as in turbulent or transient flows, a fixed wavelet basis has less structure to exploit, which may account for the narrower margin between GWCS and SGAE observed on these datasets.

As future work, we plan to incorporate the temporal dimension of PDE signals into the MLIS and NIGWT mechanisms, and to explore downstream SciML tasks performed directly in the sparse graph wavelet domain.

6 Conclusion We introduced Graph Wavelet Compressed Sensing (GWCS), a learning-based framework for compressing graph-structured scientific data, inspired by compressed sensing theory. Its core contributions are the nonparametric Multilevel Importance Sampling (MLIS) module and the scale-aware Neural Inverse Graph Wavelet Transform (NIGWT) recovery model. For controlled comparison, we additionally introduced a Sparse Graph Autoencoder (SGAE), a fully trainable, wavelet-free baseline. GWCS supports both transductive and inductive settings. It can compress an entire dataset for storage or transmission, or provide interpretable sparse representations for downstream machine learning tasks. Experimental results demonstrate the effectiveness of GWCS in compressing and reconstructing signals compared to both classical graph signal sampling and recent learning-based neural architectures.

Acknowledgments. This work was supported by the DOE SEA-CROGS project (DE-SC0023191), AFOSR project (FA9550-24-1-0231). We also thank the computing resources provided by the High Performance Computing (HPC) facility at NJIT.

References

[1] B. Adcock, A. C. Hansen, C. Poon, and B. Roman, *Breaking the coherence barrier: A new theory for compressed sensing*, in Forum of mathematics, sigma, vol. 5, Cambridge University Press, 2017, p. e4.

[2] B. Adcock, A. C. Hansen, and B. Roman, *The quest for optimal sampling: Computationally efficient, structure-exploiting measurements for compressed sensing*, in Compressed Sensing and its Applications: MATHEON Workshop 2013, Springer, 2015, pp. 143–167.

[3] A. Anis, A. Gadde, and A. Ortega, *Efficient sampling set selection for bandlimited graph signals using graph spectral proxies*, IEEE Transactions on Signal Processing, 64 (2016), pp. 3775–3789.

[4] A. Anis, A. Gadde, and A. Ortega, *Efficient sampling set selection for bandlimited graph signals using graph spectral proxies*, IEEE Transactions on Signal Processing, 64 (2016), pp. 3775–3789.

[5] Y. Bai, F. Wang, G. Cheung, Y. Nakatsukasa, and W. Gao, *Fast graph sampling set selection using gershgorin disc alignment*, IEEE Transactions on signal processing, 68 (2020), pp. 2419–2434.

[6] G. Baldan, Q. Liu, A. Guardone, and N. Thuerey, *Physics vs distributions: Pareto optimal flow matching with physics constraints*, in The Fourteenth International Conference on Learning Representations, 2026.

[7] A.-L. Barabási and R. Albert, *Emergence of scaling in random networks*, science, 286 (1999), pp. 509–512.

[8] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al., *Relational inductive biases, deep learning, and graph networks*, arXiv preprint arXiv:1806.01261, (2018).

[9] E. J. Candès, J. Romberg, and T. Tao, *Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information*, IEEE Transactions on information theory, 52 (2006), pp. 489–509.

[10] G.-H. Chen, J. Tang, and S. Leng, *Prior image constrained compressed sensing (piccs): a method to accurately reconstruct dynamic ct images from highly undersampled projection data sets*, Medical physics, 35 (2008), pp. 660–663.

[11] S. Chen, Y. C. Eldar, and L. Zhao, *Graph unrolling networks: Interpretable neural networks for graph signal denoising*, IEEE Transactions on Signal Processing, 69 (2021), pp. 3699–3713.

[12] S. Chen, M. Li, and Y. Zhang, *Sampling and recovery of graph signals based on graph neural networks*, arXiv preprint arXiv:2011.01412, (2020).

[13] S. Chen, R. Varma, A. Sandryhaila, and J. Kovačević, *Discrete signal processing on graphs: Sampling theory<? pub _ newline=""?*, IEEE transactions on signal processing, 63 (2015), pp. 6510–6523.

[14] B. Deng, *Thgsp: A pytorch-based graph signal processing library*. https://github.com/bwdeng20/thgsp, 2021.

[15] D. L. Donoho, *Compressed sensing*, IEEE Transactions on information theory, 52 (2006), pp. 1289–1306.

[16] P. ERDdS and A. R&wi, *On random graphs i*, Publ. math. debrecen, 6 (1959), p. 18.

[17] H. Gao and S. Ji, *Graph u-nets*, in international conference on machine learning, PMLR, 2019, pp. 2083–2092.

[18] W. Hamilton, Z. Ying, and J. Leskovec, *Inductive representation learning on large graphs*, Advances in neural information processing systems, 30 (2017).

[19] D. K. Hammond, P. Vandergheynst, and R. Gribonval, *Wavelets on graphs via spectral graph theory*, Applied and Computational Harmonic Analysis, 30 (2011), pp. 129–150.

[20] T. N. Kipf and M. Welling, *Variational graph auto-encoders*, arXiv preprint arXiv:1611.07308, (2016).

[21] T. N. Kipf and M. Welling, *Semi-supervised classification with graph convolutional networks*, in International Conference on Learning Representations (ICLR), 2017.

[22] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, and A. Anandkumar, *Neural operator: Learning maps between function spaces with applications to pdes*, Journal of Machine Learning Research, 24 (2023), pp. 1–97.

[23] R. Leary, Z. Saghi, P. A. Midgley, and D. J. Holland, *Compressed sensing electron tomography*, Ultramicroscopy, 131 (2013), pp. 70–91.

[24] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, *Learning nonlinear operators*

via deeponet based on the universal approximation theorem of operators, Nature machine intelligence, 3 (2021), pp. 218–229.

[25] M. Lustig, D. Donoho, and J. M. Pauly, *Sparse mri: The application of compressed sensing for rapid mr imaging*, Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine, 58 (2007), pp. 1182–1195.

[26] A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, *Sampling of graph signals with successive local aggregations*, IEEE Transactions on Signal Processing, 64 (2015), pp. 1832–1843.

[27] A. Nouranizadeh, M. Matinkia, M. Rahmati, and R. Safabakhsh, *Maximum entropy weighted independent set pooling for graph neural networks*, arXiv preprint arXiv:2107.01410, (2021).

[28] R. Ohana, M. McCabe, L. Meyer, R. Morel, F. Agocs, M. Beneitez, M. Berger, B. Burkhart, S. Dalziel, D. Fielding, et al., *The well: a large-scale collection of diverse physics simulations for machine learning*, Advances in Neural Information Processing Systems, 37 (2024), pp. 44989–45037.

[29] A. Ortega, *Introduction to graph signal processing*, Cambridge University Press, 2022.

[30] G. Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, *Random sampling of bandlimited signals on graphs*, Applied and Computational Harmonic Analysis, 44 (2018), pp. 446–475.

[31] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, Journal of Computational physics, 378 (2019), pp. 686–707.

[32] B. Ricaud, D. I. Shuman, and P. Vandergheynst, *On the sparsity of wavelet coefficients for signals on graphs*, in Wavelets and Sparsity XV, vol. 8858, SPIE, 2013, pp. 422–428.

[33] B. Roman, A. Hansen, and B. Adcock, *On asymptotic structure in compressed sensing*, arXiv preprint arXiv:1406.4178, (2014).

[34] A. Sakiyama, Y. Tanaka, T. Tanaka, and A. Ortega, *Eigendecomposition-free sampling set selection for graph signals*, IEEE Transactions on Signal Processing, 67 (2019), pp. 2679–2692.

[35] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, *Learning to simulate complex physics with graph networks*, in International conference on machine learning, PMLR, 2020, pp. 8459–8468.

[36] K. Schneider and O. V. Vasilyev, *Wavelet methods in computational fluid dynamics*, Annual review of fluid mechanics, 42 (2010), pp. 473–503.

[37] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, *The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains*, IEEE signal processing magazine, 30 (2013), pp. 83–98.

[38] N. Tremblay, P.-O. Amblard, and S. Barthelmé, *Graph sampling with determinantal processes*, in 2017 25th European signal processing conference (EUSIPCO), IEEE, 2017, pp. 1674–1678.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, *Attention is all you need*, Advances in neural information processing systems, 30 (2017).

[40] Z. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, *Hierarchical graph representation learning with differentiable pooling*, Advances in neural information processing systems, 31 (2018).

A Chebyshev polynomial approximation of SGWT. Concretely, we map the spectral interval $[\lambda_{\min}, \lambda_{\max}]$ affinely to $[-1, 1]$ (with $a = (\lambda_{\max} - \lambda_{\min})/2$, $b = (\lambda_{\max} + \lambda_{\min})/2$) and approximate each kernel by a truncated Chebyshev series

$$p(\lambda) = \tfrac{1}{2}c_0 + \sum_{k=1}^{K} c_k\, T_k\Big(\frac{\lambda - b}{a}\Big),$$

where $T_0(y) = 1$, $T_1(y) = y$ and $T_{k+1}(y) = 2yT_k(y) - T_{k-1}(y)$. Replacing λ by the matrix L (via the shifted operator $\tilde{L} = (L - bI)/a$) yields the node-domain filter $p(L)$. The action of $p(L)$ on a signal x is evaluated by the stable recurrence applied to vectors:

$$T_0(L)x = x, \qquad T_1(L)x = \tilde{L}x,$$
$$T_{k+1}(L)x = 2\tilde{L}\,T_k(L)x - T_{k-1}(L)x,$$

and accumulating

$$p(L)x = \tfrac{1}{2}c_0 x + \sum_{k=1}^{K} c_k\, T_k(L)x.$$

This avoids eigendecomposition and requires only repeated sparse matrix–vector products with $\tilde{L}$.

The Chebyshev coefficients $\{c_k\}$ are obtained by projecting the kernel onto the Chebyshev basis (equivalently via a discrete-cosine–type sampling/projection on standard Chebyshev nodes); coefficients can be computed up to a chosen degree and then truncated, so no per-degree optimization is required.

Finally, if K is the chosen polynomial degree and $|\mathcal{E}|$ the number of graph edges, the dominant cost is K sparse mat–vecs, i.e. $O(K|\mathcal{E}|)$ (plus $O(N\sum_j M_j)$ to form all filter outputs when degrees M_j differ across scales); memory overhead is modest (recurrence needs only a few N-vectors, practically $O(N)$).

B Synthetic data generating process Graph Generation. We generate synthetic datasets using three random graph models with varying structural properties. For grid graphs, we construct 2D lattice structures where each node is connected to its immediate neighbors, resulting in graphs with N nodes arranged in rectangular grids. For Erdős–Rényi graphs, we sample from $\mathcal{G}(N,p)$ where each possible edge is included independently with probability $p \in [0.01, 0.05]$, producing graphs with relatively homogeneous degree distributions. For Barabási–Albert graphs, we use preferential attachment starting from a small initial graph, where each new node connects to $m \in \{3,4,5\}$ existing nodes with probability proportional to their degrees, resulting in scale-free networks with power-law degree distributions.

For each graph type, we generate $3,000$ instances with the number of nodes N uniformly sampled from $[100, 625]$, providing diversity in graph sizes and topological characteristics.

Signal generation. For each generated graph, the signal generation process proceeds as follows. First, we compute the normalized Laplacian matrix $\mathbf{L}_{\text{norm}} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ and obtain its complete eigendecomposition $\mathbf{L}_{\text{norm}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N]$ are the eigenvectors ordered by increasing eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N \leq 2$. The eigenvectors corresponding to smaller eigenvalues represent smooth, slowly-varying patterns on the graph (low frequencies), while those associated with larger eigenvalues capture rapidly oscillating patterns (high frequencies).

The bandwidth parameter R is randomly sampled from the uniform distribution $R \sim \text{Uniform}\{3, \lfloor 0.2N \rfloor\}$, ensuring signals are predominantly low-frequency while maintaining sufficient diversity across samples. We then extract the first R eigenvectors to form the matrix $\mathbf{U}_R = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_R] \in \mathbb{R}^{N\times R}$. The spectral coefficient vector $\mathbf{a} \in \mathbb{R}^R$ is sampled independently from a Gaussian distribution $\mathbf{a} \sim \mathcal{N}(\mu\mathbf{1}, \sigma^2\mathbf{I})$, where the mean is fixed at $\mu = 0$ and the standard deviation σ is uniformly sampled from the range $[1.0, 2.0]$. This generates the clean band-limited signal as $\mathbf{x}^{\text{clean}} = \mathbf{U}_R\mathbf{a}$, which by construction lies in the span of the first R eigenvectors and thus exhibits smooth variation aligned with the graph structure.

To introduce realistic perturbations, we add Gaussian noise $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \sigma_n^2\mathbf{I})$ to obtain the observed signal $\mathbf{x} = \mathbf{x}^{\text{clean}} + \mathbf{n}$. The noise standard deviation is computed as $\sigma_n = \alpha \cdot \sigma$, where the noise level parameter α is uniformly sampled from $[0.05, 0.2]$. This ensures that the noise magnitude scales proportionally with the signal's coefficient variance, maintaining interpretable signal-to-noise ratios across different samples.

The resulting signal $\mathbf{x}$ has most of its energy in the low-frequency subspace spanned by $\mathbf{U}_R$, with additional high-frequency components contributed by the noise, making it approximately band-limited.

C Hyperparameter optimization for graph signal sampling baselines. To ensure a fair comparison between GSS methods and our learning-based approach, we conduct comprehensive hyperparameter optimization for each GSS method on the validation split of the ABL dataset. Since GSS methods are non-parametric and operate on individual graph signals without training, we optimize their hyperparameters using grid search to ensure their best possible performance.

Hyperparameter grids. For each GSS method, we define a method-specific hyperparameter grid and evaluate all combinations on the validation set. The hyperparameters are selected to minimize the mean squared error (MSE) between the reconstructed signal and the observed (noisy) signal. The hyperparameter spaces are as follows:

- **ESS**: greedy selection parameter $k \in \{1,2,3,4,5\}$, block size $\in \{1,2,3\}$, and bandwidth mode $\in \{\text{exact}, 0.1, 0.2, 0.5, 1.0, 1.5, 2.0, 2.5, 5.0\}$.
- **BSGDA**: regularization parameter $\mu \in \{10^{-4}, 10^{-3}, 10^{-2}, 0.1, 0.5, 1.0, 2.0\}$, convergence threshold $\epsilon \in \{10^{-6}, 10^{-5}, 10^{-4}\}$, localization hops $p \in \{3,6,9,12,15,21\}$, and regularization order $\in \{1,2,3\}$.
- **FastGSSS**: bandwidth mode $\in \{\text{exact}, 0.1, 0.2, 0.5, 1.0, 1.5, 2.0, 2.5, 5.0\}$, heat kernel width $\nu \in \{25, 50, 75, 100, 150\}$, Chebyshev order $\in \{8, 10, 12, 15, 20\}$, and reconstruction order $\in \{1,2,3,4,5\}$.
- **RSBS**: bandwidth mode $\in \{\text{exact}, 0.75, 1.0, 1.25, 1.5\}$, number of random vectors for eigenvalue estimation $\in \{1,3,5\}$, binary search precision $\epsilon \in \{10^{-3}, 10^{-2}, 10^{-1}\}$,

Chebyshev order $\in \{20, 30, 40\}$, reconstruction regularization $\mu \in \{10^{-3}, 10^{-2}, 0.1, 0.5\}$, regularization order $\in \{1, 2\}$, and Laplacian type $\in \{\text{combinatorial}, \text{normalized}\}$.

Bandwidth mode: Oracle vs. Data-Driven A critical hyperparameter across GSS methods is the **bandwidth mode**, which controls how much oracle information about the true signal bandwidth R is provided to the method:

- **Oracle setting (bandwidth = 'exact')**: The method receives the true bandwidth R directly. This represents an optimistic baseline and is the setting we use in our main comparisons to favor GSS methods.
- **Data-driven setting (bandwidth = $\alpha \in \mathbb{R}_+$)**: The method receives an estimated bandwidth $k = \max(2, \lfloor \alpha \cdot M \rfloor)$ where M is the sampling budget. This represents a more realistic scenario where bandwidth must be inferred from problem constraints.

In our main experiments, we report results using the oracle 'exact' bandwidth mode to provide GSS methods with maximal advantage.

D Supplementary Quantitative Metrics. Tables D.1 and D.2 report MSE and relative ℓ_2 error for all neural models. Classical GSS baselines are omitted from these tables; their RMSE comparison is the primary metric reported in Table 4.2 of the main text.

MSE results (Table D.1). MSE is related to RMSE by $\text{MSE}^{(j)} = \|\hat{\mathbf{x}}^{(j)} - \mathbf{x}^{(j)}\|_2^2/N$, so the dataset ordering is identical to RMSE. On Viscoelastic Instability, GWCS achieves 3.80×10^{-6}, more than $4\times$ lower than the next-best method (SGAE-50K, 1.88×10^{-5}), directly quantifying the squared advantage of the wavelet prior on spectrally structured data. On Turbulent Radiative Layer, SGAE-99K achieves the best MSE (27.90), but its standard deviation ($\pm$29.39) exceeds its mean, revealing severe per-sample variation: SGAE-99K occasionally fails catastrophically on individual turbulent snapshots while still achieving the best aggregate error. GWCS shows much lower variance (51.26 ± 15.49), making it more predictable across samples. On Kolmogorov Flow, SGAE-50K is best (0.0861) with GWCS (0.1019) close behind; on Dynamic Stall, SGAE-99K dominates (6.70) while GWCS (85.82) is at GXN level (90.88).

Relative ℓ_2 error (Table D.2). The relative ℓ_2 error $\text{relL2}^{(j)} = \|\hat{\mathbf{x}}^{(j)} - \mathbf{x}^{(j)}\|_2/\|\mathbf{x}^{(j)}\|_2$ is scale-invariant; values below 1.0 indicate error smaller than the signal norm.

On Viscoelastic Instability, GWCS achieves the best relative ℓ_2 (0.390), confirming that the wavelet prior captures the largest fraction of signal energy. The SGAE variants show high relative ℓ_2 (0.75–0.92) with very large standard deviations (SGAE-50K: $\pm$0.94); this is a numerical artifact, some Viscoelastic test samples have near-zero signal norm, inflating the denominator. MSE (Table D.1) is the more reliable metric for this dataset.

On Turbulent Radiative Layer and Kolmogorov Flow, the ordering matches RMSE exactly: SGAE-99K leads Turbulent (0.085) and SGAE-50K leads Kolmogorov (0.306), with GWCS close behind in both cases.

On Dynamic Stall, all relative ℓ_2 values appear small in absolute terms (0.025–0.095) because the pressure signal has a large non-zero mean ($\approx$98,160 Pa) that dominates the norm, making the denominator large. The relative ordering is nonetheless preserved (SGAE-99K best at 0.025, GWCS and GXN both near 0.092–0.095), consistent with the RMSE ranking.

Table D.1: MSE (mean $\pm$ std, physical units^2) on four PDE datasets at compression ratio $c = 5\%$. **Bold**: best; underline: second-best.

Method	Viscoelastic	Turbulent	Kolmogorov	Dynamic Stall
GXN	$4.21\times10^{-5}\pm5.02\times10^{-5}$	1724.1 ± 124.8	0.737 ± 0.190	90.88 ± 34.92
SGAE-50K	$1.88\times10^{-5}\pm2.53\times10^{-5}$	40.06 ± 43.60	$\mathbf{0.0861 \pm 0.0377}$	10.98 ± 5.97
SGAE-99K	$2.28\times10^{-5}\pm3.17\times10^{-5}$	$\mathbf{27.90 \pm 29.39}$	0.110 ± 0.038	$\mathbf{6.70 \pm 4.23}$
GWCS	$\mathbf{3.80\times10^{-6} \pm 4.46\times10^{-6}}$	51.26 ± 15.49	0.102 ± 0.026	85.82 ± 30.91

Table D.2: Relative ℓ_2 error (mean $\pm$ std) on four PDE datasets at compression ratio $c = 5\%$. **Bold**: best; underline: second-best.

Method	Viscoelastic	Turbulent	Kolmogorov	Dynamic Stall
GXN	0.947 ± 0.039	0.737 ± 0.018	0.910 ± 0.018	0.095 ± 0.020
SGAE-50K	0.923 ± 0.936	0.102 ± 0.050	$\mathbf{0.306 \pm 0.032}$	0.033 ± 0.009
SGAE-99K	0.753 ± 0.336	$\mathbf{0.085 \pm 0.042}$	0.357 ± 0.081	$\mathbf{0.025 \pm 0.008}$
GWCS	$\mathbf{0.390 \pm 0.240}$	0.126 ± 0.021	0.339 ± 0.013	0.092 ± 0.018

E Qualitative Reconstruction. Figures E.1–E.4 visualize the SGWT coefficients, MLIS sparsification mask, and NIGWT reconstruction for four representative ABL test samples over grid and Erdős–Rényi graphs at target compression ratio $c = 5\%$. Figures E.6–E.8 show qualitative reconstructions of single test sample for Turbulent Radiative Layer, Kolmogorov Flow, Dynamic Stall, and Viscoelastic Instability, respectively. Each row shows (left to right) the ground-truth signal, the reconstructed signal, and the pointwise absolute error. Neural methods (GWCS, SGAE-99K, GXN) are shown for all four datasets; classical baselines (FastGSSS, BSGDA, RSBS) are shown for Turbulent, Kolmogorov, and Dynamic Stall only, as classical evaluation of the Viscoelastic dataset was not performed due to prohibitive

per-sample computational cost in terms of time and memory.

F Code Availability The code is available at the GitHub repository: https://github.com/amrhssn/gwcs/.

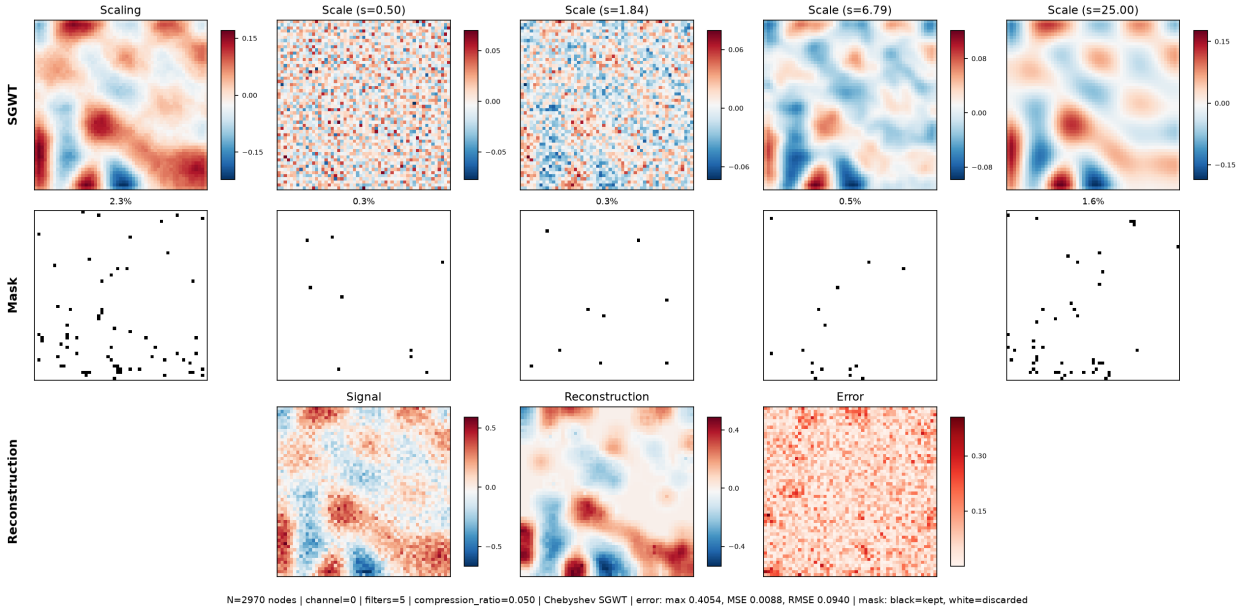


Figure E.1: GWCS pipeline visualization on a representative grid-graph ABL signal (test sample 407, $N = 2970$ nodes, RMSE$=$ 0.0940): SGWT decomposition (top), MLIS sparsification mask per scale (middle), and NIGWT reconstruction with pointwise error (bottom).

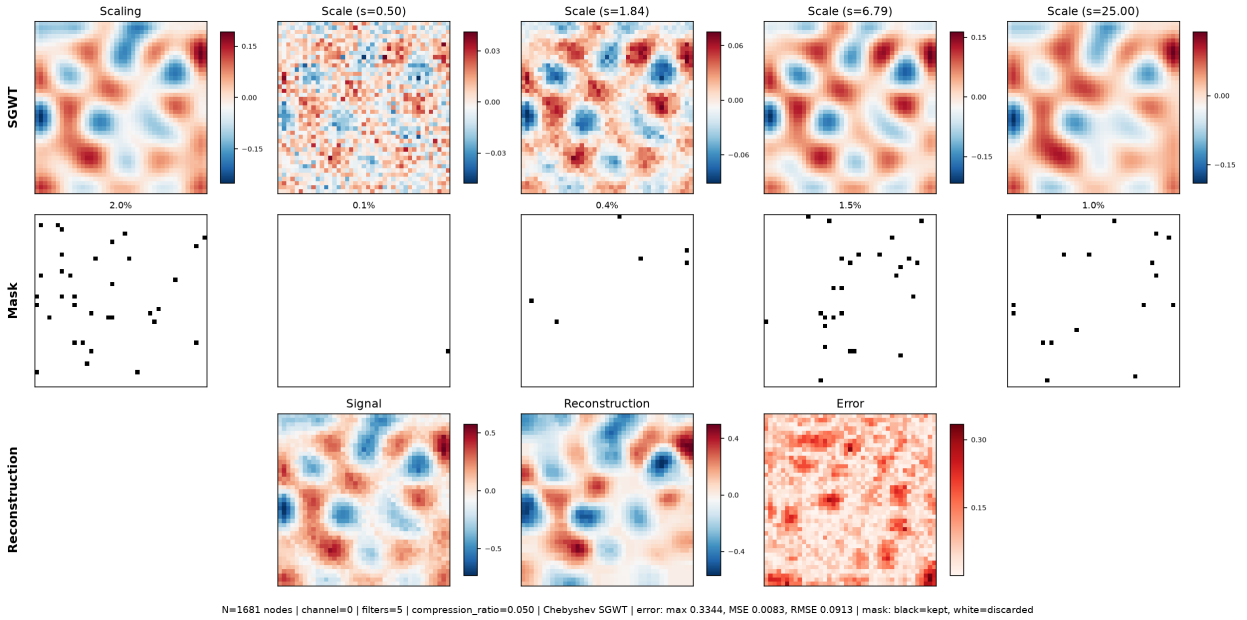


Figure E.2: GWCS pipeline visualization on a representative grid-graph ABL signal (test sample 461, $N = 1681$ nodes, RMSE$=$ 0.0913): SGWT decomposition (top), MLIS sparsification mask per scale (middle), and NIGWT reconstruction with pointwise error (bottom).

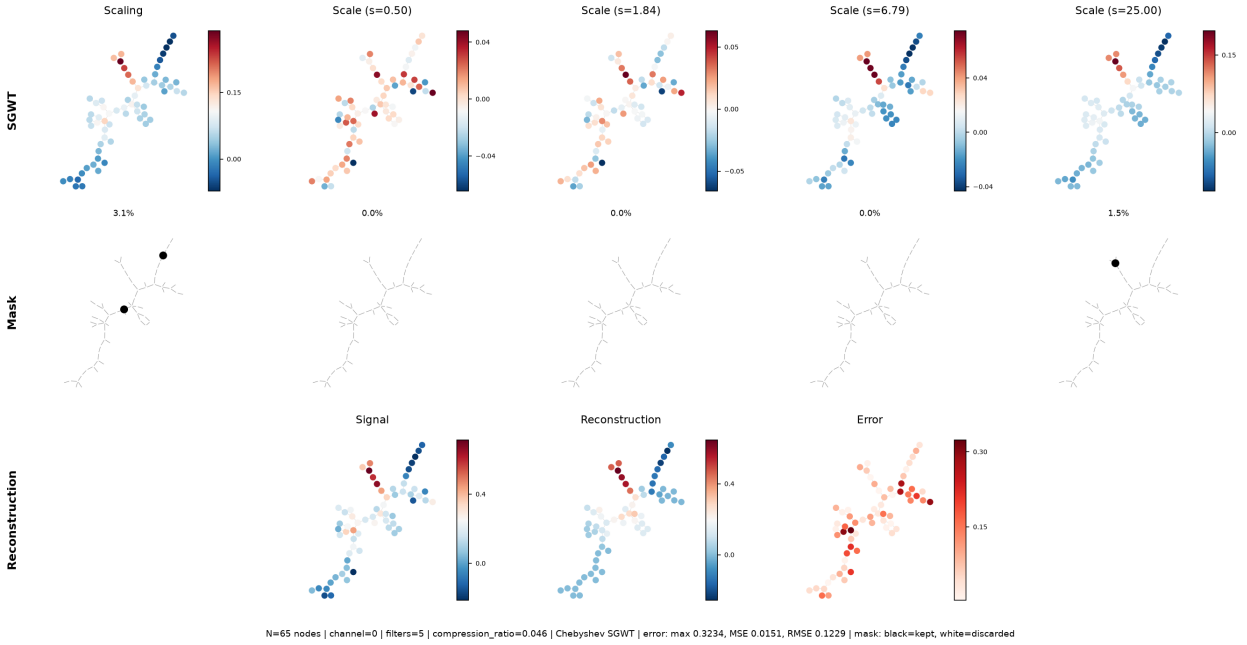


Figure E.3: GWCS pipeline visualization on a representative Erdős–Rényi-graph ABL signal (test sample 107, $N = 65$ nodes, RMSE$= 0.1229$): SGWT decomposition (top), MLIS sparsification mask per scale (middle), and NIGWT reconstruction with pointwise error (bottom).

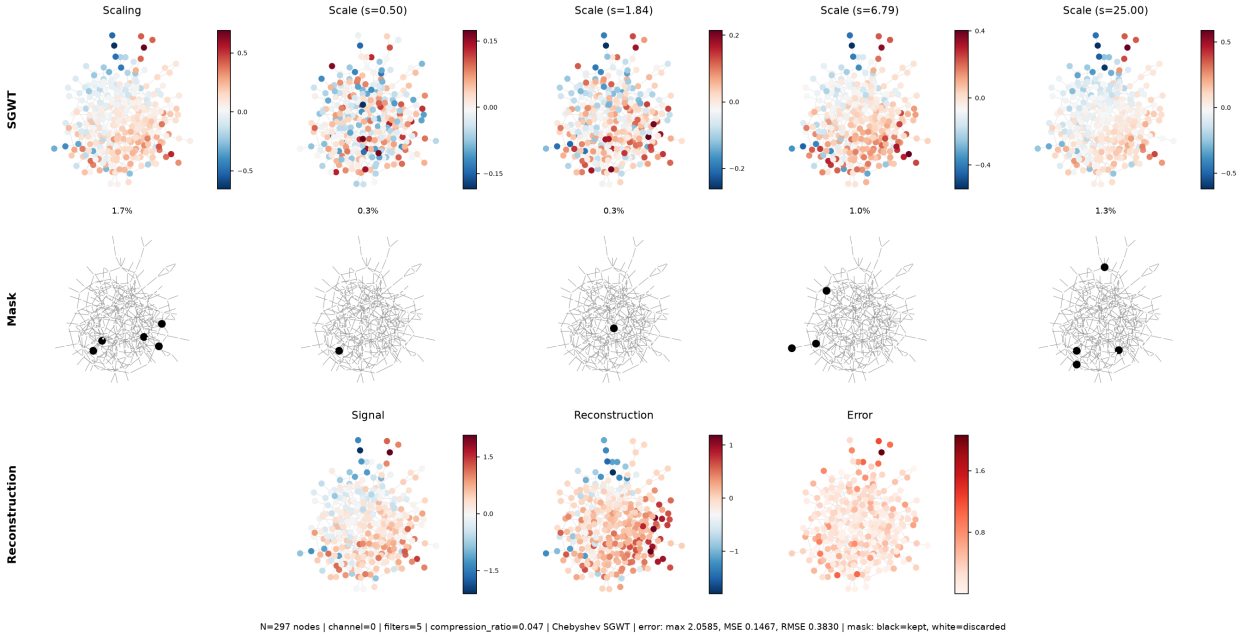
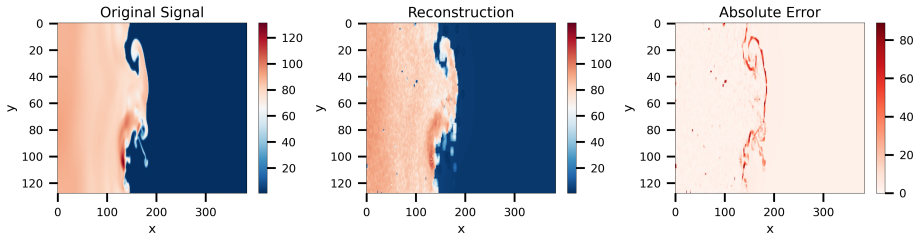
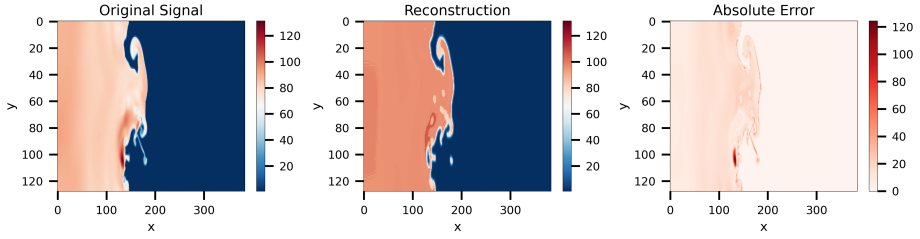


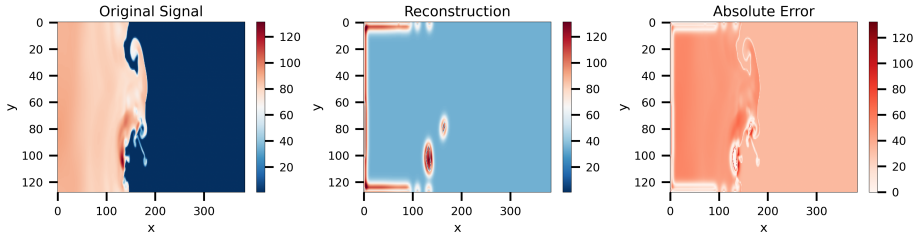
Figure E.4: GWCS pipeline visualization on a representative Erdős–Rényi-graph ABL signal (test sample 261, $N = 297$ nodes, RMSE$= 0.3830$): SGWT decomposition (top), MLIS sparsification mask per scale (middle), and NIGWT reconstruction with pointwise error (bottom).



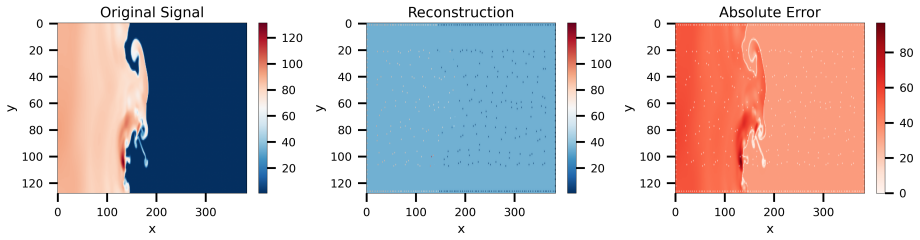
(a) GWCS (Ours) — RMSE = 6.90



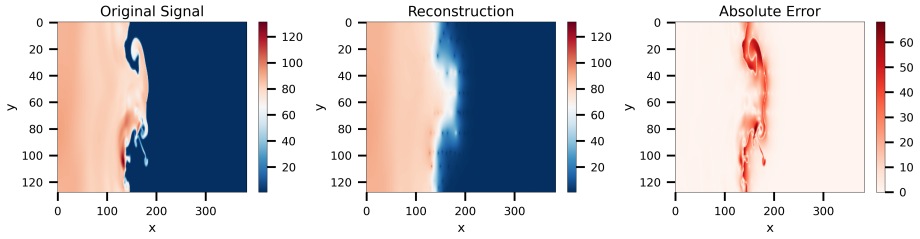
(b) SGAE-99K — RMSE = 9.32



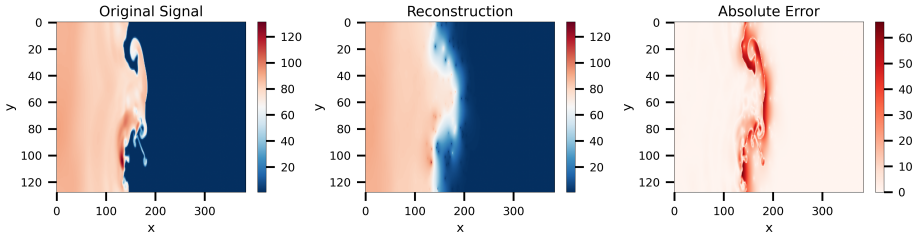
(c) GXN — RMSE = 38.55



(d) FastGSSS — RMSE = 40.47 (sample 48)

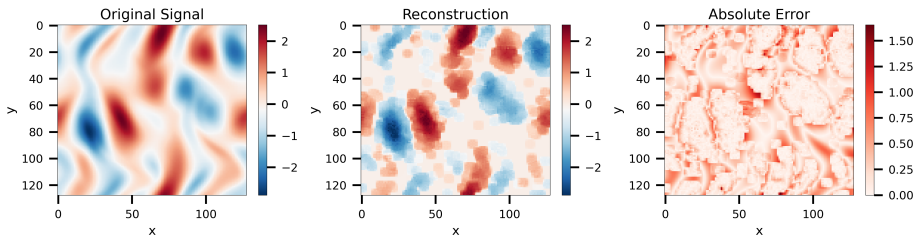


(e) BSGDA — RMSE = 8.88 (sample 48)

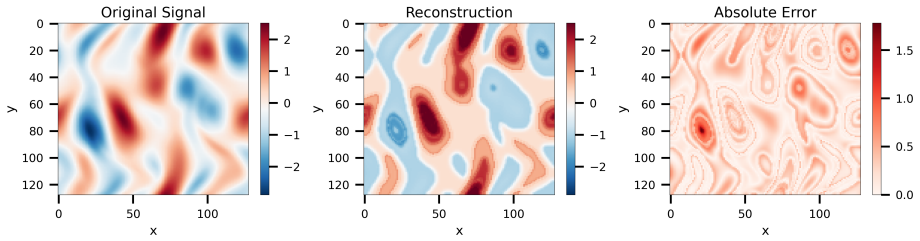


(f) RSBS — RMSE = 9.25 (sample 48)

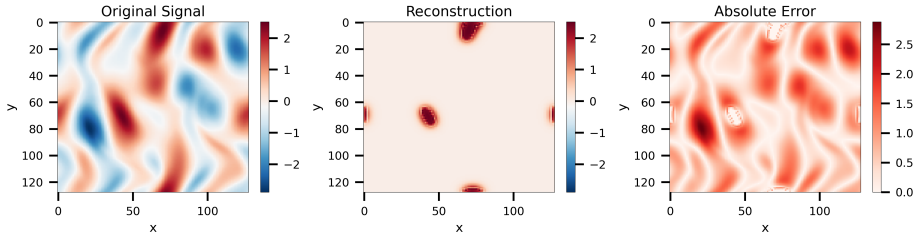
Figure E.5: Qualitative reconstruction of the Turbulent Radiative Layer density field (test sample 47).



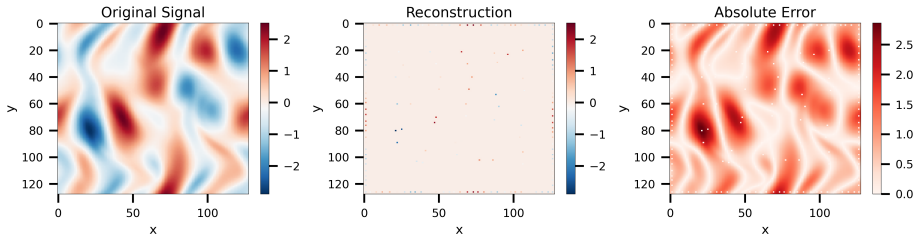
(a) GWCS (Ours) — RMSE = 0.319



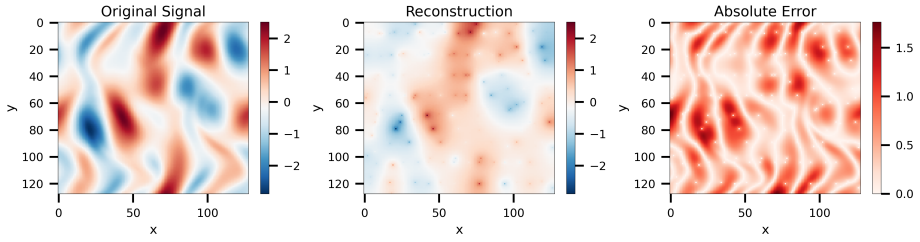
(b) SGAE-99K — RMSE = 0.328



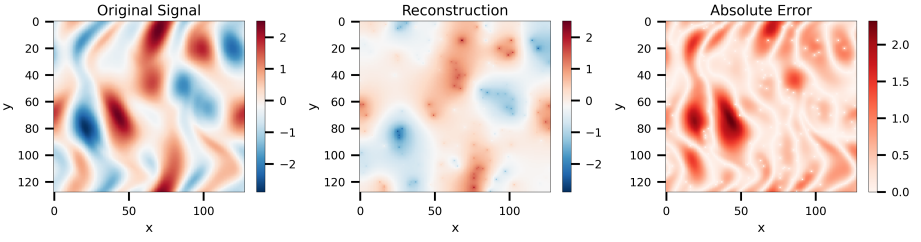
(c) GXN — RMSE = 0.851



(d) FastGSSS — RMSE = 0.903

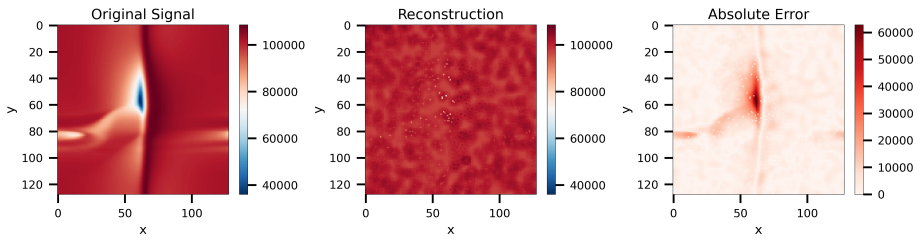


(e) BSGDA — RMSE = 0.545

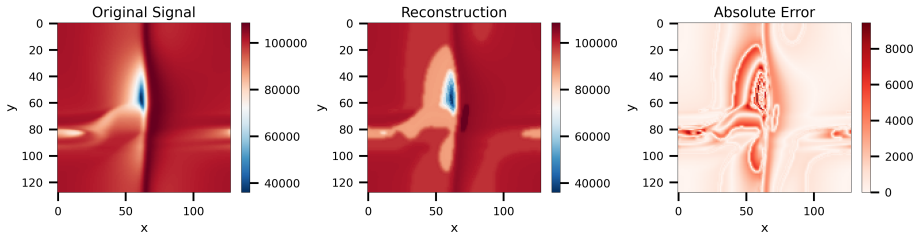


(f) RSBS — RMSE = 0.609

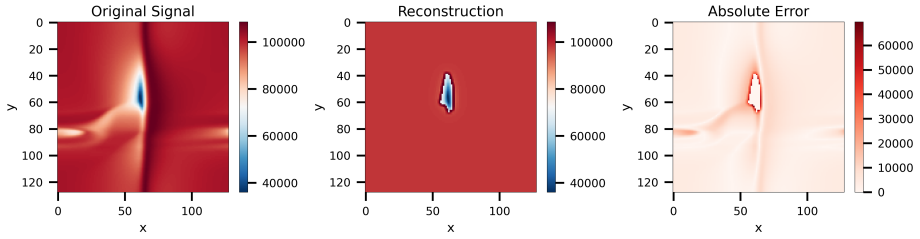
Figure E.6: Qualitative reconstruction of the Kolmogorov Flow velocity$_x$ field (test sample 47).



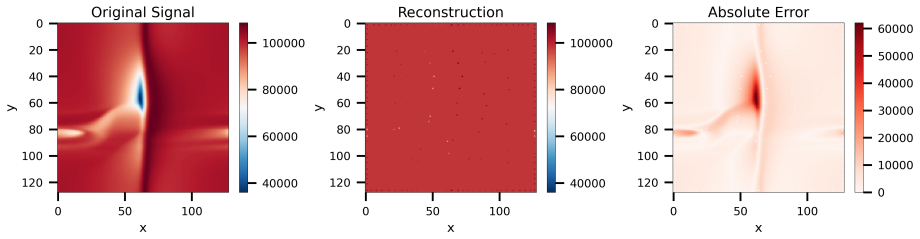
(a) GWCS (Ours) — RMSE = 9119



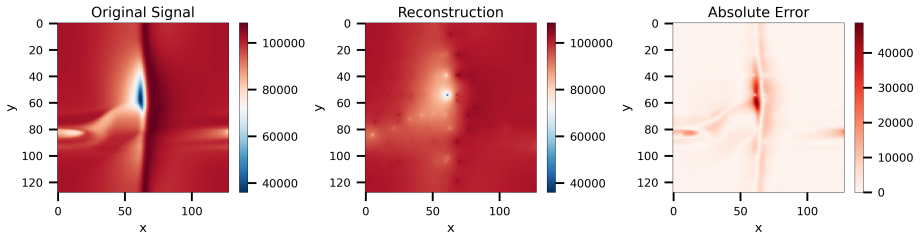
(b) SGAE-99K — RMSE = 2480



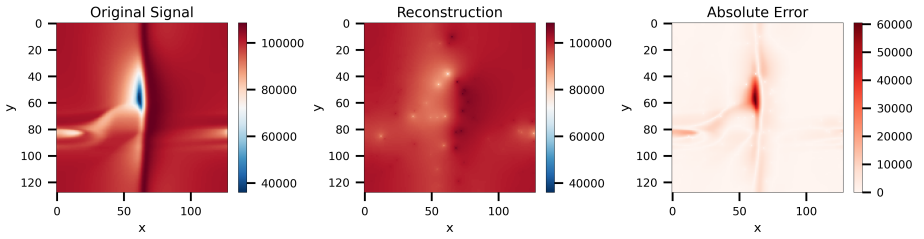
(c) GXN — RMSE = 9350



(d) FastGSSS — RMSE = 6788 (sample 48)

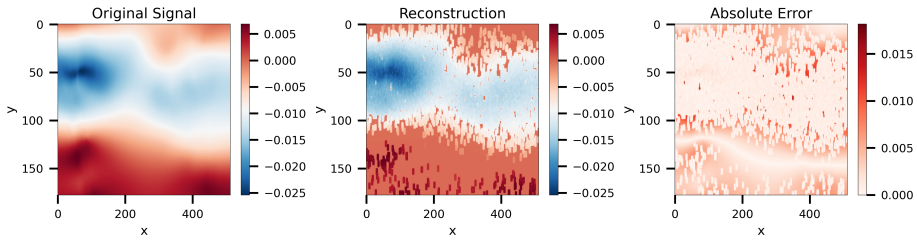


(e) BSGDA — RMSE = 3954 (sample 48)

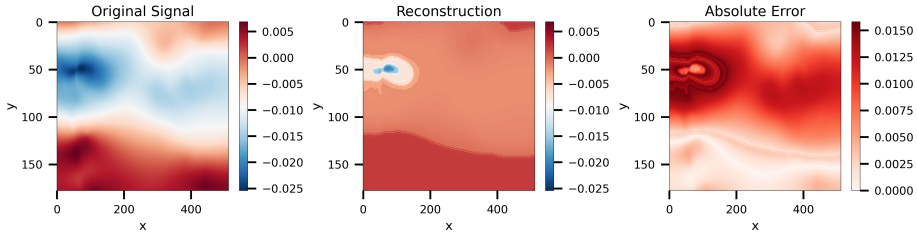


(f) RSBS — RMSE = 5033 (sample 48)

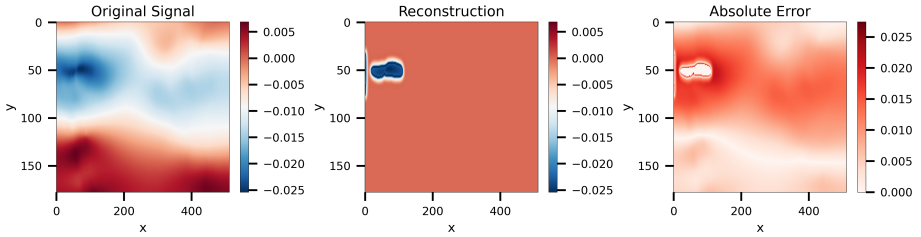
Figure E.7: Qualitative reconstruction of the Dynamic Stall pressure field (test sample 47).



(a) GWCS (Ours) — RMSE = 0.0016



(b) SGAE-99K — RMSE = 0.0037



(c) GXN — RMSE = 0.0052

Figure E.8: Qualitative reconstruction of the Viscoelastic Instability pressure field (test sample 47).