dortmund university

Zero-Shot Denoising of Distributed Acoustic Sensing Data using Deep Priors

Jannik Mänzer

Bachelor of Science

March 2025

Supervisors:
Prof. Dr. Stefan Harmeling
Sebastian Konietzny

Artificial Intelligence (VIII)
Department of Computer Science
TU Dortmund University

Abstract

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Contents

Chapter 1

Introduction

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Chapter 2

Background

This chapter provides the necessary background for the denoising methods explored in this work. We begin by defining the general denoising problem and discussing its inherent challenges. We then introduce distributed acoustic sensing (DAS) as a real-world application and highlight the unique difficulties it poses. Finally, we present key deep learning concepts and techniques that are used in the context of denoising.

2.1 Denoising

In general, denoising refers to the process of recovering a clean signal from a noisy observation. This is commonly framed as an inverse problem (IP), formally described by

$$y = \mathcal{A}(x) + n, \tag{2.1}$$

where y is the noisy observation, x is the underlying clean signal $\mathcal{A}$ refers to any forward operator and n represents some form of noise, for example Gaussian noise $n \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$. In denoising problems, $\mathcal{A}$ is simply the identity operator. Since both the noise and its distribution are often unknown, denoising is an inherently ill-posed problem, as multiple solutions can explain the same noisy data. Therefore, additional assumptions or constraints on the solution space are necessary, a process typically referred to as regularization. This can involve the use of a *prior*, which encodes our beliefs about the likely properties of the clean signal. The choice of regularizer or prior depends on the specific problem setting and the type of data involved.

2.2 Distributed Acoustic Sensing

Distributed acoustic sensing (DAS), also known as distributed vibration sensing (DVS) [10], is an innovative technology for high-resolution vibration measurements over large distances by utilizing fiber optic cables as sensor arrays. When a short laser pulse is sent through the fiber by a DAS interrogator unit, a fraction of the light is scattered back due to small variations or imperfections in the fiber. This phenomenon is referred to as Rayleigh scattering.

Vibrations along the cable caused by external influences, e.g., seismic events, strain the fiber, which in turn causes phase shifts in the backscattered light. These shifts are detected by the interrogator and, since the travel time of the light is known, can be used to accurately locate the strain along the cable [23]. In order to extract meaningful measurements, strain is analyzed over sections of the fiber, rather than at individual points. The length of these sections is called the gauge length, while another parameter, the channel spacing, determines how much this section is moved for each measurement, or channel, along the cable [8]. In practice, each channel corresponds to a virtual sensor capturing the average strain within its gauge length. Typically, the gauge length is selected to be bigger than the channel spacing, meaning that the measurement sections of neighboring channels overlap, as visualized in Figure 2.1. This concept of virtual sensors leads to high cost-effectiveness and, paired with the high sample rates enabled by the optical approach, allows measurements with significantly higher spatial and temporal resolution compared to conventional seismographs. Despite these advantages, DAS systems often suffer from much lower signal quality than conventional seismographs, as they are more sensitive to various sources of noise. These can be divided into environmental noise and optical noise.

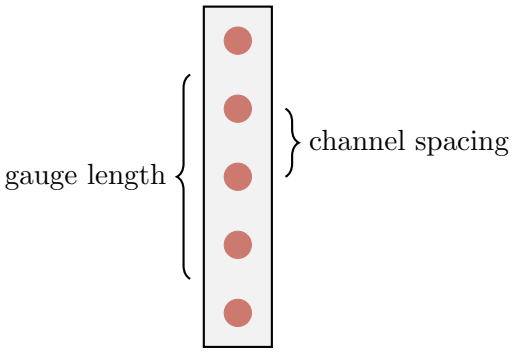

Figure 2.1: Gauge length and channel spacing. Red dots represent the individual channels along the fiber. This graphic demonstrates just one possible configuration. Figure adapted from [8].

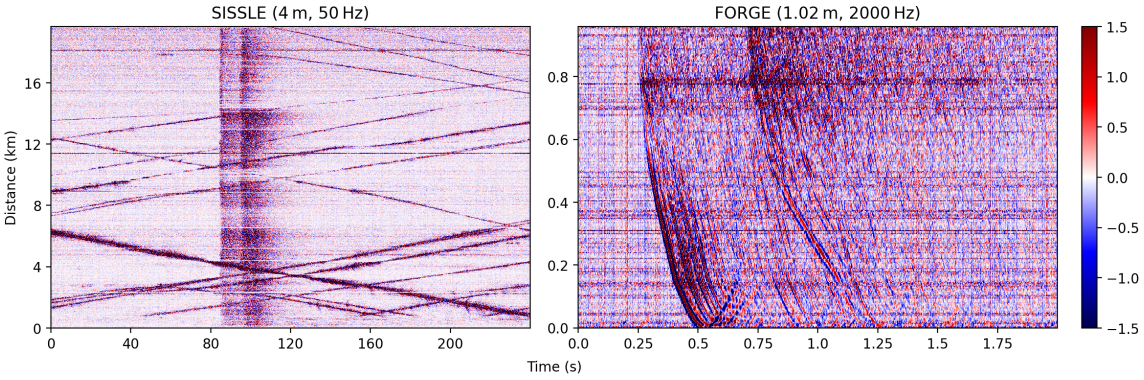

Figure 2.2: Types of noise in different DAS setups. Both measurements capture seismic activity; however, the SISSLE data mainly suffers from traffic noise (the diagonal lines). In contrast, erratic and common mode noise (the horizontal and vertical lines, respectively) are most prominent in the FORGE data. Both measurements are normalized by their respective standard deviation.

Environmental noise includes natural phenomena such as winds or ocean waves, but also vibrations caused by vehicular and pedestrian traffic. Optical noise originates from various interactions between the light and the fiber. It includes high-amplitude erratic noise and common mode noise [6]. The actual noise characteristics of DAS data not only depend on the environment, but also the measurement parameters such as channel spacing and sample rate, as visualized in Figure 2.2 for data from the SISSLE experiment near Haast, New Zealand [34] and the FORGE site in Utah [27].

2.3 Deep Learning

Deep learning is a subfield of machine learning that utilizes deep neural networks to learn complex patterns from data. Over the past decade, deep learning has established itself as the state-of-the-art approach for a wide range of problems across various different fields, such as computer vision [20], natural language processing [5] and biology [17].

2.3.1 Deep Neural Networks

In its most basic form, a neural network consists of neurons organized in layers, where each layer applies a linear transformation followed by a non-linear activation function. The output of a single layer is given by

$$y = \varphi(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{2.2}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the input, $\mathbf{W} \in \mathbb{R}^{m \times n}$ is a weight matrix, $\mathbf{b} \in \mathbb{R}^m$ is a bias vector and φ is an activation function applied element-wise, such as the Sigmoid or ReLU [36]. The outputs of each layer are then passed as inputs to the next layer, which is why this architecture is known as a fully-connected neural network. The activation functions are needed in order to avoid the network collapsing into a single linear transformation. Therefore, a neural network can be described as a function $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by θ, where θ represents the weights and biases across all layers [9]. In order to optimize these parameters, a loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is defined to measure the discrepancy between the predicted output $\hat{y} = f_\theta(x)$ and the target value y. Since $\hat{y}$ depends on θ, the loss $L(\hat{y}, y)$ is implicitly a function of θ. The gradient of the loss function with respect to the parameters, $\nabla_\theta L = \frac{\partial L}{\partial \theta}$, represents the direction of steepest ascent. Therefore, by moving the parameters in the opposite direction of the gradient, the loss function can be minimized. Typically, the gradient is not calculated for a single data point or for the whole dataset, but instead for a small subset of the dataset, which is why this approach is referred to as (mini-batch) gradient descent. Backpropagation [41] is used to efficiently compute the gradient by making use of the chain rule, enabling fast optimization. While traditionally neural networks only consisted of a few layers and required hand-crafted features to work effectively, advances in computing power allow modern architectures to automate feature extraction by using additional layers, hence the term *deep* neural network.

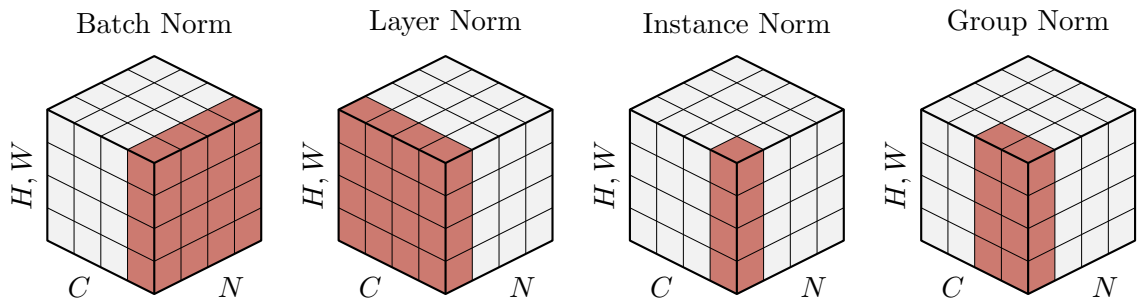

Figure 2.3: Different normalization techniques. N is the batch dimension, C is the channel dimension and H and W are the spatial dimensions of a 4D tensor. The input is normalized across the dimensions highlighted in red. Figure adapted from [50].

2.3.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) [24] are a specific type of neural network that learns features using kernels. Prior to the rise of deep learning, such kernels were designed manually for various computer vision tasks, for example the Sobel kernel [42] used for edge detection. In CNNs, these kernels are automatically learned from data. In contrast to fully-connected layers, the output of a convolutional layer is obtained by convolution with a kernel, replacing the matrix multiplication. For a kernel $\mathbf{K} \in \mathbb{R}^{m \times n}$, the convolution is defined as

$$(\mathbf{X} * \mathbf{K})_{i,j} = \sum_{k=1}^{m} \sum_{l=1}^{n} \mathbf{X}_{i+m,j+n} \cdot \mathbf{K}_{k,l}. \tag{2.3}$$

The output of the convolution is then passed through a non-linear activation function, just like in fully-connected layers. In practice, multiple kernels are used per layer, each resulting in a different feature map. These feature maps are also referred to as channels, not to be confused with the DAS channels discussed in Section 2.2.

CNNs provide two main advantages: First, since the weights are shared across the spatial dimensions, convolutional layers drastically reduce the number of parameters compared to fully-connected layers. Second, convolutions are translationally equivariant, meaning that local patterns in the input can be recognized regardless of their position, which makes CNNs very suitable for image data [9].

2.3.3 Normalization

During the training process, the inputs of each layer change with each iteration as the parameters are optimized. That slows down the training because now each layer has to adapt to the new distribution of its inputs. This process is often referred to as internal covariate shift. To counteract this issue, Ioffe et al. propose Batch Normalization (BN) [16].

The idea behind BN is to normalize the inputs across the whole mini-batch and their spatial dimensions. The normalized input for a channel i is given by

$$\hat{x}^{(i)} = \frac{x^{(i)} - \mu^{(i)}}{\sigma^{(i)}}, \tag{2.4}$$

where $\mu^{(i)}$ and $\sigma^{(i)}$ are the per-channel mean and standard deviation of the mini-batch, respectively. In order to allow the model to learn the identity if that were the optimal transformation, two additional learnable parameters, γ and β, are introduced. The output of the BN layer is then defined as

$$y^{(i)} = \gamma^{(i)}\hat{x}^{(i)} + \beta^{(i)}. \tag{2.5}$$

Since there are no batch statistics available at inference time, BN keeps track of the running mean and variance during training and uses these values for normalization. While BN is widely used, there exist various similar normalization techniques [2, 45, 50] mainly differing in the dimensions across which they are applied. A selection of them is visualized in Figure 2.3.

2.3.4 Attention Mechanisms

In neural networks, some input features are typically more important than others. An attention mechanism helps the network to focus on (attend to) the most relevant parts of the input, rather than processing all inputs equally. This works by dynamically reweighting the features based on their importance [38]. While attention is often associated with natural language processing (NLP), especially since the introduction of the Transformer architecture [46] where it is the underlying key principle, it also has applications beyond NLP. In computer vision, for example, it can help CNNs by reweighting feature channels or highlighting important spatial regions. One such method is Efficient Channel Attention (ECA), proposed by Wang et al. [48], which uses global average pooling followed by a lightweight 1D-convolution to dynamically recalibrate feature importance. For an output $\mathbf{X} \in \mathbb{R}^{C\times W\times H}$ of a convolutional layer, where W, H and C denote width, height and channel dimension, respectively, the weight of a feature channel i is given by

$$\mathbf{w}_i = \sigma(g(\mathbf{X}) * \mathbf{K})_i, \tag{2.6}$$

where σ denotes the Sigmoid activation function, $\mathbf{K} \in \mathbb{R}^{1,k}$ is a 1D-kernel with the kernel size k adaptively calculated based on C, and $g(\mathbf{x})$ represents channel-wise global average pooling:

$$g(\mathbf{X}) = \frac{1}{W \cdot H} \sum_{i=1}^{W} \sum_{j=1}^{H} \mathbf{X}_{i,j}. \tag{2.7}$$

This mechanism allows ECA to improve performance while being more efficient than predecessors like squeeze-and-excitation networks [13], which rely on more expensive fully-connected layers.

Chapter 3

Related Work

In the last few years, deep-learning-based methods have been successfully applied to various image IPs [32, 25, 15] and achieve state-of-the-art results. While these networks were traditionally trained in a supervised fashion, requiring clean target images, recent methods remove the dependency on clean data by leveraging self-supervision [53]. In this chapter, we give an overview of existing supervised and self-supervised denoising approaches, presenting key principles and discussing their limitations.

3.1 Supervised Methods

Traditional supervised methods typically employ a neural network f_θ to learn a mapping from a noisy image y to its clean counterpart x. Therefore, a dataset of paired clean and noisy images, denoted $\{(y^{(i)}, x^{(i)})\}_{i=1}^n$, is essential for the training process. The corresponding optimization problem is given by

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| f_\theta(y^{(i)}) - x^{(i)} \right\|_2^2. \tag{3.1}$$

Zhang et al. propose the denoising convolutional neural network (DnCNN) [54] which improves denoising performance by making use of residual learning, i.e., instead of directly predicting the clean image, it is trained to predict the noise in the noisy image. The denoised image is then obtained as $\hat{x} = y - f_{\theta^*}(y)$ for optimized parameters θ^*. However, depending on the problem setting, acquiring the needed clean data can be difficult or even impossible, for example in medical imaging or also in DAS.

To address this issue, Lehtinen et al. propose Noise2Noise (N2N) [26], which does not require any clean data. Instead, it utilizes two independent noisy observations $y_1 = x + n_1$ and $y_2 = x + n_2$ of the same underlying clean signal x as input and target, respectively. The training objective then becomes

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| f_\theta(y_1^{(i)}) - y_2^{(i)} \right\|_2^2. \tag{3.2}$$

This method relies on the assumption that the noise is zero-mean, i.e., $\mathbb{E}[n] = 0$, which, due to linearity of expectation, implies that $\mathbb{E}[y] = x$. Since the MSE is a mean-seeking loss function, the network then learns to predict x implicitly. Given infinite data, the optimal solution is actually equivalent to the one obtained by training with clean targets. Although N2N is often impractical in practice because the required noisy-noisy pairs are difficult to obtain, it led to the development of other self-supervised approaches.

3.2 Self-Supervised Methods

Self-supervised methods are trained similarly to traditional supervised methods, but they do not rely on externally-provided target values. In the context of denoising, these approaches can be broadly categorized into two main strategies: Noise2Noise-based methods use a training objective similar to the one given by Equation (3.2); however they generate their own noisy-noisy training pairs from individual noisy inputs. Blind-spot-based methods, on the other hand, exploit spatial correlations in the image using different masking strategies, either in the input or in the network architecture itself.

3.2.1 Noise2Noise-Based Methods

Noisier2Noise [35] builds upon N2N, but unlike N2N, it does not require a set of paired noisy-noisy images. Instead, it constructs these training pairs from individual noisy images only. Given a noisy input y, it generates an even noisier image $z = y + m = x + n + m$, with additional independent noise m following the same distribution as n. Once again, it is optimized through Equation (3.2), using z as the input and y as the target. The authors argue that $\mathbb{E}[y|z] = x + \frac{n+m}{2}$, since $\mathbb{E}[n] = \mathbb{E}[m]$. Therefore, by the same reasoning as in N2N, given a sufficient amount of noisy images, the network should learn to predict the mean of x and z, which can then be used to obtain the denoised estimate as $\hat{x} = 2f_{\theta^*}(z) - z$. While this method removes the need for a paired dataset, it requires knowledge of the noise distribution in order to sample the additional noise, which often is not available in an unsupervised setting.

Another approach based on N2N is Neighbor2Neighbor [14]. The key idea behind this method is to construct training pairs from the noisy input y by leveraging spatial redundancy through a subsampling strategy: y is divided into 2×2 cells from each of which two neighboring pixels are randomly selected — one pixel is assigned to the first subsampled image and the other to the second. These subsampled images then build the noisy training pair. As a result of the subsampling, unlike in N2N, the underlying clean signal x is not exactly identical in the two noisy images. To address this, the authors extend the training strategy given by Equation (3.2) by using an additional regularization term that encourages minimizing differences between subsampled versions of the denoised estimate.

Zero-Shot Noise2Noise [31] takes this idea one step further by enabling training on just one single noisy image instead of a set of noisy images. The term *zero-shot* refers to a training setup where the model is supposed to make predictions for types of data it has never observed before without any training examples. This approach employs a similar subsampling strategy to obtain input and target values. In order to avoid overfitting to the noisy target, it makes use of residual learning, a symmetric loss and an additional regularization term enforcing consistency with respect to the order in which downsampling and inference are performed.

3.2.2 Blind-Spot-Based Methods

All blind-spot-based methods assume that noise is zero-mean and spatially independent, while the clean image signal exhibits spatial correlations. The underlying key principle for all of them is that a network should predict the value of a given pixel in the denoised image without directly observing its noisy counterpart, hence the term *blind-spot*. Therefore, the network can only learn from the neighboring pixels, which — under the assumption of independent noise — do not carry any information about the noise affecting the target pixel, thus preventing the network from predicting a noisy image.

Krull et al. first introduce this concept in their Noise2Void (N2V) paper [21]. The authors consider training a network to predict the center pixel of a single patch of the input image in a supervised fashion, using the actual pixel value as the target. To prevent the network from simply learning the identity, they propose restricting the output pixel's receptive field by masking the center pixel. The receptive field refers to the set of pixels in the input that influences a particular pixel in the output, as visualized in Figure 3.1. However, this process is not feasible in practice, as a whole patch has to be processed to obtain a single output pixel. In order to allow efficient training, they approximate this behavior by training on random patches, for each of which a fixed number of pixels are randomly replaced by local neighbors, using their respective original noisy values as targets.

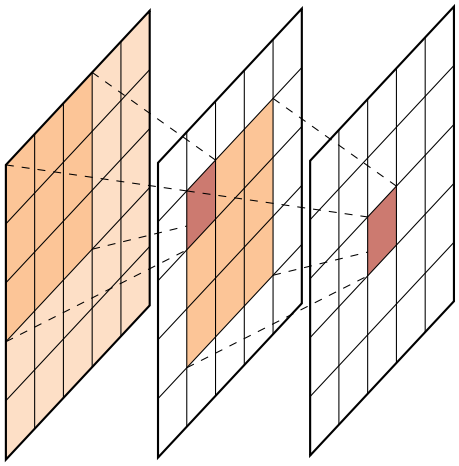

Figure 3.1: Receptive field in CNNs, using a 3×3 kernel.

In Noise2Self (N2S) [3], Batson et al. generalize this concept to sets of variables, instead of single pixels only, by introducing the notion of $\mathcal{J}$-invariance. For a noisy input image $y \in \mathbb{R}^m$, let $\mathcal{J}$ be a partition of the dimensions $\{1, \ldots, m\}$. For a subset of the dimensions $J \in \mathcal{J}$, x_J denotes x restricted to the dimensions J. A function $f : \mathbb{R}^m \to \mathbb{R}^m$

is said to be $\mathcal{J}$-invariant if, for each $J \in \mathcal{J}$, $f(x)_J$ does not depend on x_J, which implies that

$$f(x)_J = f(x_{J^c})_J, \tag{3.3}$$

where J^c refers to the complement of J. The training objective from Equation (3.1) than becomes

$$\underset{\theta}{\operatorname{argmin}} \sum_{i=1}^{n} \left\| f_\theta(y^{(i)}_{J^c})_J - y^{(i)}_J \right\|_2^2. \tag{3.4}$$

As in N2V, x_{J^c} is obtained through a masking strategy; the main difference lies in the way the masked pixels are replaced. While N2S directly uses random values, N2V chooses the replacement pixels randomly from local neighbors.

In Noise2Same [51], the authors demonstrate that, in practice, both N2V and N2S are not strictly $\mathcal{J}$-invariant and thus conclude that strict $\mathcal{J}$-invariance is not necessary for achieving good denoising performance. Therefore, they propose to do without explicit manipulation of the receptive field and instead add a regularization term that encourages the network to learn an approximately $\mathcal{J}$-invariant mapping by itself. Laine et al. [22] choose a different approach; instead of relying on masking strategies, they directly manipulate the receptive field by adapting the network architecture itself.

Chapter 4

Methods

While the self-supervised denoising methods discussed so far eliminate the need for clean data, most still require large datasets of noisy images or are limited to specific noise types. A notable exception is the Deep Image Prior (DIP) introduced by Ulyanov et al. [44]. DIP is a zero-shot method, meaning it operates on a single noisy sample, and does not make any explicit assumptions about the noise distribution.
In this chapter, we present DIP and its various extensions. First, we describe the fundamental principles of DIP, followed by common techniques used to further regularize the solution. Finally, we explore additional DIP-based approaches that build upon these foundations.

4.1 Deep Image Prior

As discussed in Section 2.1, denoising is an ill-posed inverse problem and therefore appropriate regularization is crucial in order to obtain plausible solutions. This is typically expressed as an optimization problem of the form

$$\hat{x} = \underset{x}{\operatorname{argmin}}\, L(x, y) + R(x), \tag{4.1}$$

where $L(x, y)$ is a data fidelity term that ensures that the denoised estimate $\hat{x}$ stays close to the noisy signal y and $R(x)$ is the regularizer. Most traditional, non-deep-learning methods, such as total variation denoising [40], rely on an explicit regularization term. The self-supervised methods discussed in Section 3.2 do not necessarily include such a term; however, they do make specific assumptions about the solution (e.g., zero-mean or spatially independent noise), which are implicitly encoded in the training procedure — through techniques like subsampling or masking.
In contrast, DIP does not make any explicit assumptions about the noise or image structure. Instead, it relies solely on the architecture of a convolutional neural network to implicitly regularize the solution. The key idea is to parameterize an image x as the output of an CNN f_θ through $x = f_\theta(z)$, where z refers to a random vector, e.g., $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

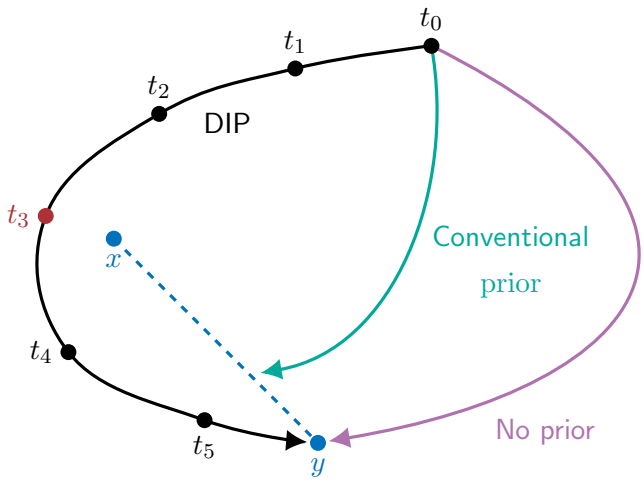

Figure 4.1: The effect of priors. Without regularization, the optimization leads directly to the noisy image y. Conventional priors shift the solution closer to the clean image x. DIP will eventually overfit to y, but often the optimization path will pass close to x, with the optimal stopping point marked in red. Figure adapted from [44].

This means that instead of optimizing x directly, the reconstruction is constrained by the network's ability to map z to a plausible image. In terms of Equation (4.1), the MSE is used as the loss function and the regularizer $R(x)$ is replaced with the implicit prior induced by the network structure, leading to the following optimization:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \|f_\theta(z) - y\|_2^2 . \tag{4.2}$$

After training the network using gradient descent, the denoised estimate is then obtained as $\hat{x} = f_{\theta^*}(z)$. The random vector z remains fixed throughout the training process. The regularizing effect of this parameterization relies on the observation that CNNs tend to capture structured patterns (e.g., edges and textures) before fitting to high-frequency noise. However, since the network is sufficiently expressive, it will eventually memorize the noise, leading to overfitting. Thus, it is crucial to identify the optimal stopping point, at which the network successfully learned the underlying image structure while minimizing the influence of noise. This process is visualized in Figure 4.1. Formally, this regularizer can be expressed as

$$R(x) = \begin{cases} 0 & f_\theta \text{ can produce } x \text{ in } N \text{ steps} \\ +\infty & \text{else,} \end{cases} \tag{4.3}$$

where N refers to a fixed maximum number of iterations. However, such a fixed stopping point is not ideal because the optimal stopping point depends on factors such as the specific image x, the random vector z and also the initial network parameters θ_0.

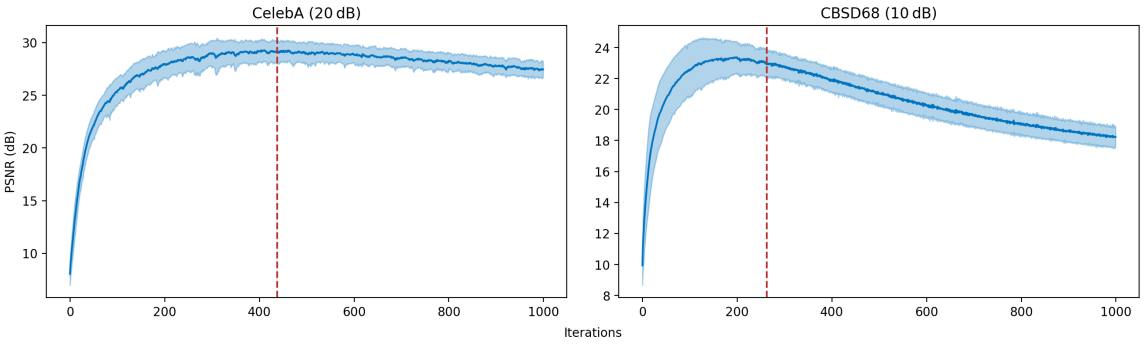

Figure 4.2: Early stopping via windowed moving variance on different datasets and noise levels. Image quality is evaluated on 20 random images from the CelebA [29] (left) and CBSD68 [33] (right) datasets, with Gaussian noise added at specified levels. The blue line represents the mean PSNR, the light blue shaded area indicates the standard deviation. The red line marks the average detected stopping point.

4.1.1 Early Stopping

The process of halting the training to mitigate overfitting is known as early stopping (ES). As discussed earlier, a fixed stopping point generally will not work well. Therefore, we need to find a way to dynamically detect optimal stopping points during training. In principle, if the clean ground-truth image x were available, this would be trivial: One could simply track image quality using an appropriate metric, e.g., PSNR or SSIM, and stop training at its peak. However, since x is inherently unknown in the denoising setting, we need an alternative criterion to determine when to stop.

Wang et al. [47] propose such a criterion based on the running variance of the DIP reconstructions over time. Let $\{x_t\}_{t\geq 1}$ denote the sequence of the respective denoised estimates $x_t = f_{\theta_t}(z)$ at iteration t. The authors observe that the MSE $\|x_t - x\|_F^2$ initially drops as the networks learns the image structure, and then rises again due to overfitting to noise, leading to a U-shaped curve. However, once again, x is unknown in practice, so the goal is to detect the minimum of said curve without access to x. To achieve this, they consider the running variance over a window of W iterations, given by

$$\mathrm{Var}(t) = \frac{1}{W}\sum_{w=0}^{W-1}\left\| x_{t+w} - \frac{1}{W}\sum_{i=0}^{W-1} x_{t+i}\right\|_F^2. \tag{4.4}$$

Intuitively, when t is near the optimal stopping point, x_t should be close to x, leading to $\frac{1}{W}\sum_{w=0}^{W} x_t \approx x$. Plugging this back into Equation (4.4), we see that when t is near the optimum, $\mathrm{Var}(t)$ approximates the average MSE across the window. Therefore, they propose using the minimum of the variance curve to determine the stopping point. To improve robustness, they introduce a patience parameter P, allowing the variance to stagnate for up to P iterations before stopping. This approach, termed early stopping via windowed moving variance (ES-WMV), is effective across different noise levels and types of images, as demonstrated in Figure 4.2.

4.1.2 Total Variation

Another way to prevent overfitting is to incorporate an additional explicit regularization term, such as total variation (TV) [40]. TV encourages piecewise smoothness by penalizing abrupt intensity changes in the image. Formally, for an image $x \in \mathbb{R}^{W \times H}$, it is defined as

$$\mathrm{TV}(x) = \sum_{i=1}^{W-1} \sum_{j=1}^{H-1} (|x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}|). \tag{4.5}$$

Liu et al. propose combining the implicit network regularization of DIP with an explicit TV regularizer, leading to the method known as DIP-TV [28]. This results in the following optimization problem:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \, \|f_\theta(z) - y\|_2^2 + \lambda \mathrm{TV}(f_\theta(z)), \tag{4.6}$$

where λ is a hyperparameter controlling the strength of the TV regularization.

4.2 Deep Diffusion Image Prior

Diffusion models [43] have emerged as a powerful class of generative models, achieving state-of-the-art performance in various applications, including image synthesis, denoising, and inverse problems. The key idea is to transform complex data distributions into simple ones (such as Gaussian noise) via a forward stochastic process and then train a model to approximate the reverse process. Formally, given data $x_0 \sim q(x_0)$, the forward diffusion process produces a sequence of increasingly noisy samples x_t over time $t \in [0, T]$ using a fixed Markovian process:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t \mathbf{I}), \tag{4.7}$$

where β_t is a predefined variance schedule controlling how much noise is added at each step. For sufficiently large T, x_T approximates pure Gaussian noise. A key property of this process is that it admits a closed-form solution at arbitrary timesteps:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1 - \bar{\alpha}_t)\mathbf{I}), \tag{4.8}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t} (a - \beta_s)$ is the cumulative noise factor. Using the reparameterization trick [18], this can be expressed as

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{4.9}$$

In DDPMs [11], a simple linear schedule for β_t is used. In IDDPM [37] an improved cosine noise schedule is introduced, which better balances noise levels across timesteps and leads to higher sample quality. The goal of the diffusion model is to learn the reverse process, parameterized by a neural network, which iteratively removes noise:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \beta_t \mathbf{I}), \tag{4.10}$$

where μ_θ is a neural network predicting the mean of the denoised sample.

Chung et al. combine DIP with ideas from diffusion models, leading to the Deep Diffusion Image Prior (DDIP) [7]. They observe that both DIP and diffusion models aim to recover a posterior mean: DIP estimates $\mathbb{E}[x|z,y]$ and diffusion models estimate $\mathbb{E}[x_0|x_t,y]$. As both z and x_t are distributed according to $\mathcal{N}(\mathbf{0},\mathbf{I})$ for $t=T$, they propose a generalization of DIP to multiple noise scales. Instead of optimizing from pure noise, they iteratively reduce the noise in z, gradually steering it toward the clean image x_0.

$$\text{for } t=T,\dots,1: \theta_{t-1} = \underset{\theta_t}{\operatorname{argmin}} \left\| f_{\theta_t}(x_t) - y \right\|_2^2, \tag{4.11}$$

$$x_{t-1} = \sqrt{\bar{\alpha}_t} f_{\theta_{t-1}}(x_t) + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0},\mathbf{I}). \tag{4.12}$$

In practice, DDIP uses a pre-trained diffusion model as f_θ and adapts the underlying prior distribution using LoRA [12] — optimizing only a small set of additional parameters. However, since this work focuses strictly on zero-shot methods, we consider using a completely untrained network, as $f_\theta(x_t)$ can still be seen as an estimate of x_0 after a sufficient number of initial iterations.

4.3 Self-Guided Deep Image Prior

Another extension of DIP is the Self-Guided Deep Image Prior (SG-DIP) [4]. Similar to DDIP, it also iteratively updates the network input; however, instead of doing so explicitly, it achieves this implicitly by optimizing both the network's parameters and the input itself. Formally, the proposed training objective is defined as:

$$\theta^*, z^* = \underset{\theta,z}{\operatorname{argmin}} \left\| \mathbb{E}_n[f_\theta(z+n)] - y \right\|_2^2 + \lambda \left\| \mathbb{E}_n[f_\theta(z+n)] - z \right\|_2^2, \tag{4.13}$$

where y refers to the noisy image, z is a random tensor, n is Gaussian noise sampled as $n \sim \mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})$ and λ is a hyperparameter balancing the two terms. Empirically, the authors find that setting $\sigma = \frac{\max(z)}{2}$ yields good results. The first term enforces data fidelity, while the second term acts as a regularization term, encouraging the network to behave as a denoiser by constraining the output to be close to the learned input z, thereby mitigating overfitting. The expectation over the noise distribution helps to stabilize the learned mapping; the denoised estimate is then obtained as $\hat{x} = \mathbb{E}_n[f_\theta(z+n)]$. In practice, the expectation is approximated by averaging the outputs for multiple noise samples.
This objective naturally induces a structured learning process: Ideally, the network first learns to capture the general structure of the image. Due to the optimization of z and the regularization term, both the input and output should provide a reasonable estimate of the clean image. At this stage, vanilla DIP would typically begin to overfit. However, in SG-DIP, the regularization term constrains the output from deviating too much from the input, making it costly for the network to fit the noise — since doing so would introduce global alterations to the output. Instead, the network is encouraged to refine the output while staying close to the input, effectively suppressing noise without overfitting.

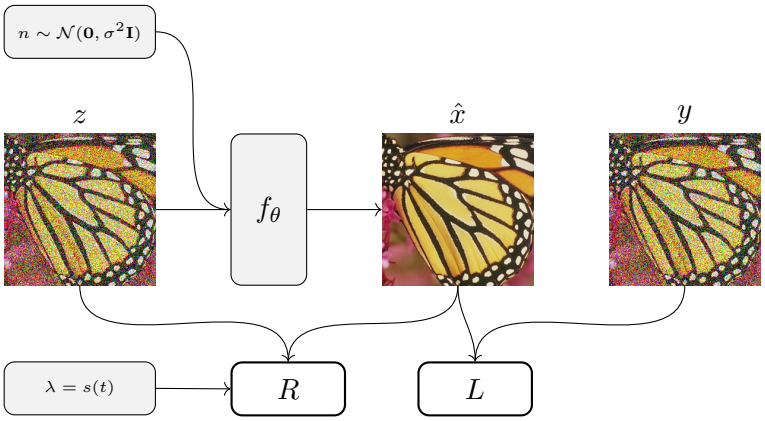

Figure 4.3: Overview of our adapted SGR-DIP training objective. The network input is the sum of z (which equals y in the first iteration) and noise sampled from a Gaussian distribution. The overall loss consists of two terms: a data fidelity term L and a regularization term R. L measures the difference between the network output and the noisy image, R ensures that the output stays close to the input. The influence of R increases over time through a schedule $s(t)$.

To explicitly reinforce this behavior, we propose gradually increasing λ throughout training according to a schedule $\lambda = s(t)$, where t denotes the current iteration. This approach intuitively divides the optimization process into two phases: Initially, both terms in the objective are balanced, allowing the network to extract the general structure of the image. As training progresses and λ increases, the regularization term dominates, restricting the model to only minor refinements.

We observe that the addition of the schedule improves denoising performance; however, we note that the effectiveness of this approach is highly dependent on the specific configuration of the schedule implemented. If the regularization term remains too small for an extended period, there is still a risk of overfitting. Conversely, if it increases too prematurely, it may hinder further learning before the image structure has been adequately captured. To address this issue, we also consider a variant in which we replace the random initialization of z with the noisy image y. Note that the two loss terms are still different, as z is optimized throughout training, while y stays constant. This modification, which we term Self-Guided Refinement Deep Image Prior (SGR-DIP), facilitates the network's ability to learn the primary structure of the image. However, we observe that it necessitates an increase in the variance of the noise added to z to prevent the network from merely learning the identity function. The overall training objective is illustrated in Figure 4.3.

Chapter 5

Implementation Details

Building on the methods described in the previous chapter, this chapter focuses on the practical aspects of our work. Specifically, it covers the architecture design, evaluation metrics, datasets, and preprocessing steps used in our experiments.

5.1 Architecture

All architectures in our experiments use a fully-convolutional encoder-decoder network with skip connections, following the U-Net [39] architecture. The encoder progressively downsamples the input using strided convolutions, increasing the number of feature channels while reducing spatial resolution. The decoder then upsamples the feature representations back to the target resolution using bilinear upsampling followed by additional convolutions. Skip connections are employed between corresponding encoder and decoder layers, bypassing the bottleneck to preserve fine-grained details. This helps retain spatial information that would otherwise be lost due to downsampling, thereby improving reconstruction quality. We find that parameterizing these skip connections is crucial for achieving strong denoising performance. While the original DIP paper employs 1×1 convolutions in the skip connections, we also explore a variant that replaces these convolutions with ECA blocks. We find that both variants yield comparable performance in terms of PSNR and SSIM; however, ECA blocks often contribute to better preservation of fine details, as shown in Figure 5.1.

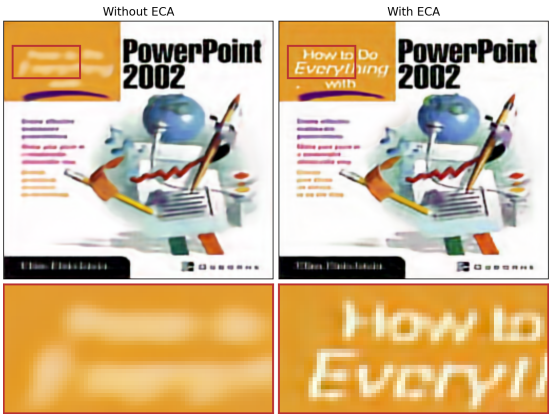

Figure 5.1: Denoised results for both skip connection variants. Both images are obtained using SG-DIP. The ECA-based variant retains more structural details compared to the convolution-based approach.

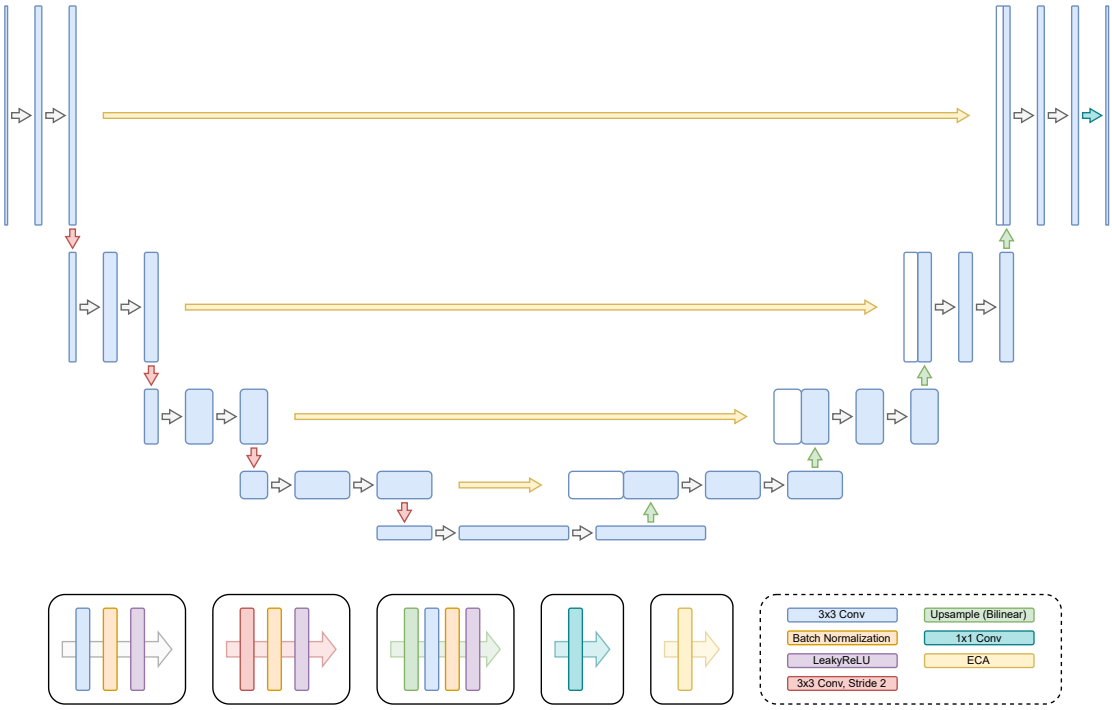

Figure 5.2: Architecture used in our experiments. Each blue box corresponds to a multi-channel feature map. White boxes represent copied feature maps, reweighted using ECA. Different colored arrows represent different operations, visualized in detail in the boxes below. Figure adapted from [39] and [44].

As in the original DIP paper, Batch Normalization is applied throughout the network and all activation functions are LeakyReLU [30]. LeakyReLU extends ReLU by allowing a small slope for negative values, ensuring non-zero gradients throughout the domain. Formally, it is defined as

$$\varphi(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha x & \text{if } x < 0, \end{cases} \tag{5.1}$$

where $\alpha = 0.2$ in our case. Unless noted otherwise, all experiments employ the architecture visualized in Figure 5.2, implemented using PyTorch [1]. Optimization is performed using the Adam optimizer [19] with a learning rate of 0.01.

5.2 Metrics

To evaluate and compare the performance of different denoising methods, we rely on two widely used image quality metrics: Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM).

5.2.1 Peak Signal-to-Noise Ratio

PSNR is a standard metric used to quantify the reconstruction quality of an image by measuring the ratio between the maximum possible pixel intensity and the mean squared error (MSE) between the original and denoised images, $x, \hat{x} \in \mathbb{R}^{W \times H}$. It is defined as:

$$\mathrm{PSNR}(x, \hat{x}) = 10 \cdot \log_{10} \left(\frac{x_{\max}^2}{\mathrm{MSE}(x, \hat{x})} \right), \tag{5.2}$$

where $x_{\max}$ is the maximum possible pixel intensity. For a normalized image, $x_{\max} = 1$, therefore we obtain the simplified definition

$$\mathrm{PSNR}(x, \hat{x}) = 10 \cdot \log_{10} \left(\frac{1}{\mathrm{MSE}} \right) = -10 \cdot \log_{10}(\mathrm{MSE}(x, \hat{x})), \tag{5.3}$$

where the MSE is given by

$$\mathrm{MSE}(x, \hat{x}) = \frac{1}{W \cdot H} \sum_{i=1}^{W} \sum_{j=1}^{H} (x_{i,j} - \hat{x}_{i,j})^2. \tag{5.4}$$

For color images with dimensions $3 \times W \times H$, the MSE is additionally computed across the 3 color channels.
A higher PSNR value indicates better image quality, with less distortion introduced by the denoising process. However, PSNR is a pixel-wise metric that does not account for perceptual quality, therefore we also consider the SSIM.

5.2.2 Structural Similarity Index Measure

SSIM [49] is designed to assess perceptual similarity between images by considering structural information, contrast, and luminance. It is computed as:

$$\mathrm{SSIM}(x, \hat{x}) = \frac{(2\mu_x \mu_{\hat{x}} + c_1)(2\sigma_{x\hat{x}} + c_2)}{(\mu_x^2 + \mu_{\hat{x}}^2 + c_1)(\sigma_x^2 + \sigma_{\hat{x}}^2 + c_2)} \tag{5.5}$$

where μ_x and $\mu_{\hat{x}}$ are the mean intensities of the original and denoised images, σ_x^2 and $\sigma_{\hat{x}}^2$ are their variances, and $\sigma_{x\hat{x}}$ is the covariance between them. The constants c_1 and c_2 stabilize the division, preventing numerical instabilities when the denominator is close to zero. SSIM values range from -1 to 1, with higher values indicating better structural similarity. Unlike PSNR, SSIM is more aligned with human perception, making it a crucial metric for evaluating image denoising performance.

5.3 Datasets

This section describes the datasets used in our experiments for both image and distributed acoustic sensing (DAS) data.

5.3.1 Image Datasets

We use three standard image datasets commonly employed for image denoising tasks:

- **Set14** [52]: A small dataset consisting of 14 images, widely used for evaluating image restoration algorithms.
- **CBSD68** [33]: A dataset containing 68 natural images from the Berkeley segmentation dataset, commonly used for benchmarking denoising methods.
- **CelebA** [29]: A large-scale face dataset with over 200,000 images, useful for evaluating denoising performance on human faces.

To generate a noisy image $\hat{x}$ from a clean image x at a predefined PSNR level, we add Gaussian noise $n \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, where σ^2 is determined by the desired PSNR. Because the noise is additive, substituting $\hat{x} = x + n$ into Equation 5.4 simplifies it to

$$\text{MSE}(x, \hat{x}) = \frac{1}{W \cdot H} \sum_{i=1}^{W} \sum_{j=1}^{H} n_{i,j}^2. \tag{5.6}$$

Since the noise is also zero-mean, we approximate $\sigma^2 \approx \text{MSE}$. Rewriting Equation 5.3 in terms of MSE, we get

$$\text{MSE}(x, \hat{x}) = 10^{-\frac{\text{PSNR}(x,\hat{x})}{10}}. \tag{5.7}$$

For a given PSNR value, we can therefore obtain a corresponding noisy image as

$$\hat{x} = x + \sigma\epsilon, \quad \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{5.8}$$

where $\sigma = \sqrt{10^{-\frac{\text{PSNR}(x,\hat{x})}{10}}}$.

5.3.2 Distributed Acoustic Sensing Datasets

For denoising DAS data, we use data from two real-world experiments:

- **SISSLE** [34]: The South Island Seismology at the Speed of Light Experiment near Haast, New Zealand.
- **FORGE** [27]: The Frontier Observatory for Research in Geothermal Energy in Utah.

Exact sources for the samples used in our experiments can be found in the supplementary material.

5.4 Preprocessing

This section discusses preprocessing steps applied to the input before passing it to DIP-based methods. For image data, no specific preprocessing is needed. After generating noisy samples as described in the last section, we simply standardize the inputs to zero mean and unit variance.

For DAS data, however, appropriate preprocessing is crucial for the success of DIP-based methods. As discussed in Section 2.2, noise in DAS measurements often exhibits a grid-like structure, mainly due to erratic and common mode noise. This structured noise is repetitive and spatially correlated, and therefore inherently easy to learn for a convolutional network, which contradicts a key assumption of DIP — that meaningful signal is learned faster than noise. To mitigate this issue, following [6], we first apply a low-pass filter with a high-cut frequency of 200 Hz. To further remove common mode noise, we subtract the time-wise median from the filtered signal.

Another challenge arises due to the large variations in amplitude within DAS measurements. DIP-based methods, particularly SG-DIP, tend to focus on high-amplitude regions while neglecting lower-amplitude parts. To address this, we propose using a local normalization approach. Specifically, we divide the input into overlapping local regions and compute the mean and standard deviation within each window. Each region is then normalized independently to have zero mean and unit variance:

$$x'_{i,j} = \frac{x_{i,j} - \mu_{i,j}}{\sigma_{i,j}}, \tag{5.9}$$

where $\mu_{i,j}$ and $\sigma_{i,j}$ are the mean and standard deviation computed within a local window centered at (i, j). In practice, we use a window size of 32×32.

After DIP-based processing, we rescale the output using the stored local mean and standard deviation to restore the original amplitude relationships. The different preprocessing steps and their corresponding denoised results are visualized in Figure **??**.

Chapter 6

Results

Chapter 7

Discussion

Chapter 8

Conclusion

Bibliography

[1] Jason Ansel et al. "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation". In: *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, Apr. 2024. DOI: 10.1145/3620665.3640366. URL: https://pytorch.org/assets/pytorch2-2.pdf.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. arXiv: 1607.06450 [stat.ML]. URL: https://arxiv.org/abs/1607.06450.

[3] Joshua Batson and Loic Royer. "Noise2Self: Blind Denoising by Self-Supervision". In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, Sept. 2019, pp. 524–533. URL: https://proceedings.mlr.press/v97/batson19a.html.

[4] Evan Bell et al. "Robust Self-Guided Deep Image Prior". In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5. DOI: 10.1109/ICASSP49357.2023.10096631.

[5] Tom B. Brown et al. "Language models are few-shot learners". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546.

[6] Yangkang Chen et al. "Denoising of Distributed Acoustic Sensing Seismic Data Using an Integrated Framework". In: *Seismological Research Letters* 94.1 (Nov. 2022), pp. 457–472.

[7] Hyungjin Chung and Jong Chul Ye. "Deep Diffusion Image Prior for Efficient OOD Adaptation in 3D Inverse Problems". In: *Computer Vision – ECCV 2024*. Ed. by Aleš Leonardis et al. Cham: Springer Nature Switzerland, 2025, pp. 432–455. ISBN: 978-3-031-73226-3.

[8] Tim Dean, T. Cuny, and Arthur Hartog. "Determination of the Optimum Gauge Length for Borehole Seismic Surveys Acquired Using Distributed Vibration Sensing". In: June 2015. DOI: 10.3997/2214-4609.201412740.

[9] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[10] Arthur H. Hartog. *An Introduction to Distributed Optical Fibre Sensors*. CRC Press, 2017. URL: http://dx.doi.org/10.1201/9781315119014.

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[12] Edward J Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=nZeVKeeFYf9.

[13] Jie Hu et al. *Squeeze-and-Excitation Networks*. 2019. arXiv: 1709.01507 [cs.CV]. URL: https://arxiv.org/abs/1709.01507.

[14] Tao Huang et al. "Neighbor2Neighbor: Self-Supervised Denoising From Single Noisy Images". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 14781–14790.

[15] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. "Globally and locally consistent image completion". In: *ACM Trans. Graph.* 36.4 (July 2017). ISSN: 0730-0301. DOI: 10.1145/3072959.3073659. URL: https://doi.org/10.1145/3072959.3073659.

[16] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *CoRR* abs/1502.03167 (2015). arXiv: 1502.03167. URL: http://arxiv.org/abs/1502.03167.

[17] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589. DOI: 10.1038/s41586-021-03819-2.

[18] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: 1312.6114 [stat.ML]. URL: https://arxiv.org/abs/1312.6114.

[19] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[21] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. "Noise2Void - Learning Denoising From Single Noisy Images". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 2124–2132. DOI: 10.1109/CVPR.2019.00223.

[22] Samuli Laine et al. "High-Quality Self-Supervised Deep Image Denoising". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/2119b8d43eafcf353e07d7cb5554170b-Paper.pdf.

[23] S Lapins et al. "DAS-N2N: machine learning distributed acoustic sensing (DAS) signal denoising without clean data". In: *Geophysical Journal International* 236.2 (Nov. 2023), pp. 1026–1041. ISSN: 1365-246X. DOI: 10.1093/gji/ggad460. URL: http://dx.doi.org/10.1093/gji/ggad460.

[24] Y. LeCun et al. "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4 (1989), pp. 541–551. DOI: 10.1162/neco.1989.1.4.541.

[25] Christian Ledig et al. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 105–114. DOI: 10.1109/CVPR.2017.19.

[26] Jaakko Lehtinen et al. "Noise2Noise: Learning Image Restoration without Clean Data". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 2965–2974. URL: https://proceedings.mlr.press/v80/lehtinen18a.html.

[27] A. Lellouch et al. "Low-Magnitude Seismicity With a Downhole Distributed Acoustic Sensing Array—Examples From the FORGE Geothermal Experiment". In: *Journal of Geophysical Research: Solid Earth* 126.1 (Dec. 2020). ISSN: 2169-9356. DOI: 10.1029/2020jb020462. URL: http://dx.doi.org/10.1029/2020JB020462.

[28] Jiaming Liu et al. "Image Restoration Using Total Variation Regularized Deep Image Prior". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2019, pp. 7715–7719. DOI: 10.1109/ICASSP.2019.8682856.

[29] Ziwei Liu et al. "Deep Learning Face Attributes in the Wild". In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 3730–3738. DOI: 10.1109/ICCV.2015.425.

[30] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. "Rectifier nonlinearities improve neural network acoustic models". In: *ICML*. Vol. 30. 1. Atlanta, GA. 2013, p. 3.

[31] Youssef Mansour and Reinhard Heckel. "Zero-Shot Noise2Noise: Efficient Image Denoising without any Data". In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2023, pp. 14018–14027. DOI: 10.1109/CVPR52729.2023.01347.

[32] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. "Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper_files/paper/2016/file/0ed9422357395a0d4879191c66f4faa2-Paper.pdf.

[33] D. Martin et al. "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics". In: *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. Vol. 2. 2001, 416–423 vol.2. DOI: 10.1109/ICCV.2001.937655.

[34] Meghan S. Miller, John Townend, and Voon Hui Lai. "The South Island Seismology at the Speed of Light Experiment (SISSLE): Distributed Acoustic Sensing Across and Along the Alpine Fault, South Westland, New Zealand". In: *Seismological Research Letters* (Dec. 2024). ISSN: 0895-0695. DOI: 10.1785/0220240322. eprint: https://pubs.geoscienceworld.org/ssa/srl/article-pdf/doi/10.1785/0220240322/7068598/srl-2024322.1.pdf. URL: https://doi.org/10.1785/0220240322.

[35] Nick Moran et al. "Noisier2Noise: Learning to Denoise From Unpaired Noisy Data". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[36] Vinod Nair and Geoffrey E. Hinton. "Rectified linear units improve restricted boltzmann machines". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Haifa, Israel: Omnipress, 2010, pp. 807–814. ISBN: 9781605589077.

[37] Alexander Quinn Nichol and Prafulla Dhariwal. "Improved Denoising Diffusion Probabilistic Models". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by Marina Meila and Tong Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 18–24 Jul 2021, pp. 8162–8171. URL: https://proceedings.mlr.press/v139/nichol21a.html.

[38] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. "A review on the attention mechanism of deep learning". In: *Neurocomputing* 452 (2021), pp. 48–62. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2021.03.091. URL: https://www.sciencedirect.com/science/article/pii/S092523122100477X.

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4.

[40] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: Nonlinear Phenomena* 60.1 (1992), pp. 259–268. ISSN: 0167-2789. DOI: https://doi.org/10.1016/0167-2789(92)90242-F. URL: https://www.sciencedirect.com/science/article/pii/016727899290242F.

[41] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323 (Oct. 1986), pp. 533–. URL: http://dx.doi.org/10.1038/323533a0.

[42] Irwin Sobel. "An Isotropic 3x3 Image Gradient Operator". In: *Presentation at Stanford AI Lab (SAIL)* (1968).

[43] Jascha Sohl-Dickstein et al. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 2256–2265. URL: https://proceedings.mlr.press/v37/sohl-dickstein15.html.

[44] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. "Deep Image Prior". In: *International Journal of Computer Vision* 128.7 (Mar. 2020), pp. 1867–1888. ISSN: 1573-1405. DOI: 10.1007/s11263-020-01303-4. URL: http://dx.doi.org/10.1007/s11263-020-01303-4.

[45] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. "Instance Normalization: The Missing Ingredient for Fast Stylization". In: *CoRR* abs/1607.08022 (2016). arXiv: 1607.08022. URL: http://arxiv.org/abs/1607.08022.

[46] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[47] Hengkang Wang et al. "Early Stopping for Deep Image Prior". In: *Transactions on Machine Learning Research* (2023). ISSN: 2835-8856. URL: https://openreview.net/forum?id=231ZzrLC8X.

[48] Qilong Wang et al. "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 11531–11539. DOI: 10.1109/CVPR42600.2020.01155.

[49] Zhou Wang et al. "Image quality assessment: from error visibility to structural similarity". In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.

[50] Yuxin Wu and Kaiming He. "Group Normalization". In: *CoRR* abs/1803.08494 (2018). arXiv: 1803.08494. URL: http://arxiv.org/abs/1803.08494.

[51] Yaochen Xie, Zhengyang Wang, and Shuiwang Ji. "Noise2Same: Optimizing A Self-Supervised Bound for Image Denoising". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 20320–20330. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/ea6b2efbdd4255a9f1b3bbc6399b58f4-Paper.pdf.

[52] Roman Zeyde, Michael Elad, and Matan Protter. "On Single Image Scale-Up Using Sparse-Representations". In: *Curves and Surfaces*. Ed. by Jean-Daniel Boissonnat et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 711–730. ISBN: 978-3-642-27413-8.

[53] Dan Zhang et al. *Unleashing the Power of Self-Supervised Image Denoising: A Comprehensive Review*. 2024. arXiv: 2308.00247 [eess.IV]. URL: https://arxiv.org/abs/2308.00247.

[54] Kai Zhang et al. "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising". In: *IEEE Transactions on Image Processing* 26.7 (July 2017), pp. 3142–3155. ISSN: 1941-0042. DOI: 10.1109/tip.2017.2662206. URL: http://dx.doi.org/10.1109/TIP.2017.2662206.

Appendix A

Supplementary Information

A.1 Acronyms

BN	batch normalization
CNN	convolutional neural network
DAS	distributed acoustic sensing
DDIP	deep diffusion image prior
DIP	deep image prior
ECA	efficient channel attention
ES	early stopping
IP	inverse problem
MSE	mean squared error
N2N	Noise2Noise
N2S	Noise2Self
N2V	Noise2Void
PSNR	peak signal-to-noise ratio
SG-DIP	self-guided deep image prior
SSIM	structural similarity index measure
TV	total variation

Table A.1: List of common acronyms.