

Self-Supervised Learning from Noisy and Incomplete Data

Julián Tachella
CNRS, ENS Lyon

Mike Davies
University of Edinburgh

January 7, 2026

Abstract

Many important problems in science and engineering involve inferring a signal from noisy and/or incomplete observations, where the observation process is known. Historically, this problem has been tackled using hand-crafted regularization (e.g., sparsity, total-variation) to obtain meaningful estimates. Recent data-driven methods often offer better solutions by directly learning a solver from examples of ground-truth signals and associated observations. However, in many real-world applications, obtaining ground-truth references for training is expensive or impossible. Self-supervised learning methods offer a promising alternative by learning a solver from measurement data alone, bypassing the need for ground-truth references. This manuscript provides a comprehensive summary of different self-supervised methods for inverse problems, with a special emphasis on their theoretical underpinnings, and presents practical applications in imaging inverse problems.

Contents

1	Introduction to self-supervised learning for inverse problems	4
1.1	Inverse problems	5
1.2	From analytic reconstruction to machine learning	6
1.3	Supervised learning	7
1.4	Self-supervised learning	9
1.5	What this manuscript is not about	11
1.6	Outline	12
2	Learning from noisy measurements	14
2.1	Learning from independent noisy pairs	14
2.2	Known noise distribution	16
2.3	Partially unknown noise distribution	23
2.4	Learning generative models from noisy data	29
2.5	Towards general inverse problems	31
2.6	Summary	31
3	Learning from incomplete measurements	33
3.1	Leveraging multiple operators	33
3.2	Leveraging invariance to transformations	39
3.3	Learning generative models from incomplete measurements	44
3.4	Model identification theory	45
3.5	Summary	48
4	Finite dataset effects	50
4.1	Hold-out method with self-supervised losses	50
4.2	Variance of the loss and its gradients	50
4.3	Gap with supervised learning	53
4.4	Fine-tuning and test-time adaptation	55
5	Extensions and open problems	56
5.1	Non-linear inverse problems	56
5.2	Towards large scale self-supervised imaging	57
5.3	Robust solutions and partially defined models	57
5.4	Beyond the ℓ_2 loss	58
5.5	Uncertainty quantification and generative modelling	59
5.6	Sample complexity	59

5.7 Choosing the right self-supervised method	60
A Noisier2Noise and R2R equivalence	62
B Identification of moments	63
C Additional proofs	65

Chapter 1

Introduction to self-supervised learning for inverse problems

Many important problems in science and engineering boil down to inferring a signal or image from noisy and/or incomplete observations, where the measurement process, often a physical system, is *a priori* known. For example, this includes the large range of applications in sensing and imaging inverse problems, from learning the structure of molecules using computational microscopy to astronomical imaging. In healthcare, medical imaging via computational tomography (CT), Magnetic resonance imaging (MRI), and ultrasound provides a crucial component of early diagnosis of disease. While applications in time series and audio include source separation, acoustic tomography and blind deconvolution.

While, historically, such inverse problems were solved through model-based approaches [1], the powerful representation learning properties of deep neural networks have allowed researchers to develop new state-of-the-art data-driven reconstructions. Such solutions, trained on large quantities of ground truth data, are able to exploit the sophisticated statistical dependencies that previous hand-crafted models, such as sparse representations or total variation (TV) regularization [2], do not capture, and have substantially raised the bar on the achievable image reconstruction performance, e.g., in accelerated MRI image reconstruction [3], showing a significant 6 dB gain in peak signal-to-noise ratio (PSNR) over TV regularization.

Despite the phenomenal success of such solutions, their reliance on large amounts of ground truth training data is a key limitation of the technology, restricting its application to problems where access to ground truth data is readily available - ones that have therefore essentially already been "solved" beforehand. This is particularly problematic in important scientific, medical and engineering settings, as well as for sensing systems working in complex environments, where ground truth data is scarce and where prediction accuracy is of overriding importance. This, in turn has led to a growing interest in the development of new *self-supervised learning* solutions that aim to learn reconstructions without direct access to ground truth data.

The goal of this monograph is to provide a self-contained presentation of such self-supervised learning techniques that have emerged within recent years and highlighting the links to the underpinning statistical and geometric theory for such methods.

1.1 Inverse problems

The main focus of all such methods is the solution of a mathematical *inverse problem* to estimate or reconstruct a signal or image of interest. While often these may in reality be defined as continuous functions, in order to compute a solution it is necessary to represent it in a discrete form, e.g., through an appropriate basis function expansion [1]. At the risk of committing an inverse crime [4] we will focus in this manuscript on discrete signals, represented/approximated as a finite dimensional vector, $\mathbf{x} \in \mathbb{R}^n$ (or $\mathbf{x} \in \mathbb{C}^n$) that can be estimated from measurements, $\mathbf{y} \in \mathbb{R}^m$, through the stable *inversion* of an acquisition process, also called the *forward operator*, $\mathbf{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$, that we assume to have already accommodated the discretization process:

$$\mathbf{y} = \mathbf{A}(\mathbf{x}) + \boldsymbol{\epsilon}. \quad (1.1)$$

Here $\boldsymbol{\epsilon}$ captures any noise or modelling errors and should be assumed to be possibly signal dependent, like the case of Poisson noise [5].

Examples

We can illustrate the forward model in (1.1) with a few idealized examples that we will use throughout this manuscript:

- **Denoising** is the simplest inverse problem, where the forward operator is the identity mapping, that is $\mathbf{A}(\mathbf{x}) = \mathbf{x}$, and the goal is to remove the noise from the observed measurements.
- **Image inpainting** consists of recovering a set of missing pixels in an image, that is $\mathbf{A}(\mathbf{x}) = \text{diag}(\mathbf{b})\mathbf{x}$ with mask $\mathbf{b} \in \{0, 1\}^n$.
- **Super resolution** is generally modelled as an inverse problem [6] with $\mathbf{A}(\mathbf{x}) = \mathbf{S} \text{circ}(\mathbf{k})\mathbf{x}$ where $\text{circ}(\mathbf{k}) \in \mathbb{R}^{n \times n}$ is a convolution with a kernel $\mathbf{k} \in \mathbb{R}^n$ and $\mathbf{S} \in \mathbb{R}^{m \times n}$ is a sub-sampling operation.
- **Accelerated magnetic resonance imaging** can be written as a linear inverse problem [3]. In the single-coil setting, the acquisitions can be modelled as $\mathbf{A}(\mathbf{x}) = \text{diag}(\mathbf{b})\mathbf{F}\mathbf{x}$ where $\mathbf{F} \in \mathbb{C}^{n \times n}$ is the 2D discrete Fourier transform and $\mathbf{b} \in \{0, 1\}^n$ is the acceleration mask.
- **Phase retrieval** is a non-linear inverse problem, which can be written as $\mathbf{A}(\mathbf{x}) = |\mathbf{B}\mathbf{x}|^2$ where $\mathbf{B} \in \mathbb{C}^{m \times n}$ is a linear operator, which can be either random or structured according to the application [7].
- **Inverse scattering** is a complex non-linear inverse problem related to the Helmholtz equation, which can be written [8] as

$$\mathbf{A}(\mathbf{x}) = \text{circ}(\mathbf{g})\text{diag}(\mathbf{x})(\mathbf{I} - \text{circ}(\mathbf{g})\text{diag}(\mathbf{x}))^{-1}\mathbf{v}$$

where $\text{circ}(\mathbf{g})$ denotes a convolution with Green's kernel $\mathbf{g} \in \mathbb{C}^n$ and $\mathbf{v} \in \mathbb{C}^n$ is the incident field.

1.2 From analytic reconstruction to machine learning

Solving an inverse problem consists in devising a reconstruction function $f : \mathbb{R}^m \mapsto \mathbb{R}^n$ which removes the noise and inverts the effect of the forward operator, such that $f(\mathbf{y}) \approx \mathbf{x}$ approximately recovers the underlying signal. In many imaging and sensing scenarios, the forward model (1.1) is linear, and early systems were explicitly designed to ensure that sufficient measurements were acquired in order that reconstruction through iterative or direct inversion $f(\mathbf{y}) = \mathbf{A}^{-1}\mathbf{y}$ could be used. However, as sensing and imaging problems became more challenging there was a need for more sophisticated reconstruction techniques.

1.2.1 Why it is hard to invert?

The key challenges in solving any ill-conditioned inverse problem are two-fold. First, the measurements acquired are generally not noise free. For example, low flux imaging results in observing only a limited number of photons at each measurement - something that is typically modeled statistically as Poisson noise. Part of the role of the inversion process is therefore to be able to infer clean signals from noisy observations.

The second major challenge is due to an inability to acquire a ‘complete’ set of measurements. Sometimes this is a result of explicit undersampling, for example, in the accelerated MRI example above. In other scenarios the level of incompleteness is more subtle, such as in deconvolution problems where the forward operator might be full rank but severely ill-posed, or in super-resolution example, where the notion of undersampling is to a certain extent user defined.

In either case, incomplete measurements means that there are insufficient measurements to simply directly invert the problem. For example, in the case of linear problems, the forward operator may be rank deficient with a non-trivial null space resulting in an infinity of possible measurement consistent solutions, or the forward operator may be full rank, but severely ill-conditioned, meaning that there will be no general *stable* inverse.

Geometrically, solving the problem of incomplete measurements requires, at least implicitly, the restriction of the signal model to a low dimensional set, as the image of a stable (i.e. Lipschitz) mapping, $f : \mathbb{R}^m \mapsto \mathbb{R}^n$, can at most have dimension m . This, for example, was the underpinning idea behind the compressed sensing revolution [9], that popularized the notion of sparse signal models.

1.2.2 Model based reconstruction

Tackling inverse problems with noise and incomplete measurements historically used statistical techniques that combined a model-based consistency loss with the addition of some statistical constraint to capture the desired properties of the signals of interest. For example, this is often achieved by solving a regularized variational optimization problem composed of an ℓ_2 consistency loss¹, $\|\mathbf{y} - \mathbf{A}(\mathbf{x})\|^2$, and a regularization term, $\rho(\mathbf{x})$, that captures the prior knowledge of the set of signals of interest:

$$f(\mathbf{y}) = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}(\mathbf{x})\|^2 + \rho(\mathbf{x}). \quad (1.2)$$

¹In settings with non-Gaussian noise, the ℓ_2 consistency is often replaced by the negative log-likelihood or other robust alternatives.

Many different regularizers have been used depending on the precise application, ranging from classical Tikhonov regularization to those that encourage sparse or low rank solutions, such as TV or nuclear norm regularization. However, such hand-crafted regularization can rarely capture all the sophisticated statistical dependencies within the problem leading researchers to explore the possibility of developing superior data-driven solutions.

1.3 Supervised learning

The standard (supervised) way of learning inverse problem solvers from data consists of using a neural network, f , as the reconstruction function, $\hat{\mathbf{x}} = f_{\boldsymbol{\theta}}(\mathbf{y})$ with weights, $\boldsymbol{\theta} \in \mathbb{R}^p$, learned directly from training data that consist of pairs of ground truth signals and their associated measurements: $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$. This is typically achieved by minimizing some supervised loss,

$$f^* = \arg \min_f \sum_{i=1}^N \mathcal{L}_{\text{SUP}}(\mathbf{x}_i, \mathbf{y}_i, f) \quad (1.3)$$

such as the ℓ_2 loss

$$\mathcal{L}_{\text{SUP}}(\mathbf{x}, \mathbf{y}, f) = \frac{1}{n} \|f(\mathbf{y}) - \mathbf{x}\|^2 \quad (1.4)$$

to give the learned solution, f^* , where we have dropped the explicit dependence on the weight vector, $\boldsymbol{\theta}$, and instead consider the optimization in the space of admissible functions.

In principle, if the class of admissible neural network functions is sufficiently flexible *and* we have sufficient training data, such an approach should allow us to approximate the optimal reconstruction function, which in the case of the ℓ_2 loss is the conditional mean estimator:

$$f^*(\mathbf{y}) \approx \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ \mathbf{x} \} \quad (1.5)$$

where the expectation² here is taken with respect to the posterior distribution $p(\mathbf{x}|\mathbf{y})$.

The learning approach transforms the problem into one of regression and potentially enables us to fully exploit the structure available within the training data. In practice, as we will briefly discuss in Section 1.5, the choice of the neural network architecture will also play an important role in the performance of the learned inverse mapping.

1.3.1 Commonly used network models

Various different neural networks configurations for the inverse mapping, f , have been proposed for inverse problem solvers. Here, we focus on the two main classes of solutions that have been considered, noting that this will inevitably be incomplete in such a rapidly evolving field.

Most imaging solutions leverage an efficient low-level vision subnetwork structure that provides a image-to-image mapping, $g : \mathbb{R}^n \mapsto \mathbb{R}^n$, and is typically realized through either ResNet [10] or UNet [11, 12] style architectures with more recent incarnations incorporating attention mechanisms, e.g. [13]. This subnetwork is then used in various ways that differ primarily in how the acquisition model, \mathbf{A} , is incorporated into the overall solution. There are two broad approaches.

²Throughout the manuscript we will use the notation $\mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ \phi(\mathbf{x}) \}$ to denote the expectation of $\phi(\mathbf{x})$ under $p(\mathbf{x}|\mathbf{y})$.

Back-projection networks A popular, simple and yet effective solution is to first map the measurements back into the image/signal domain using a back-projection operator. In the linear case, this can be done using the linear pseudo-inverse \mathbf{A}^\dagger or some easily computable surrogate to this, such as \mathbf{A}^\top . The subnetwork, $\phi(\mathbf{u})$, is then used to map the backprojected signal to the clear reconstructed one. The full reconstruction function then takes the form $f(\mathbf{y}) = \phi(\mathbf{A}^\dagger \mathbf{y})$ and is trained in an end-to-end manner.

Unrolled architectures These networks are motivated by attempting to mimic the structure of an iterative optimization algorithm unrolled for a small number of iterations, with the image-to-image mapping playing the role of a *proximal* type operator. Probably the simplest such algorithm is the proximal gradient descent variant that takes the form:

$$\mathbf{x}^{(k+1)} = \phi_k \left(\mathbf{x}^{(k)} - \tau \frac{\partial \mathbf{A}^\top}{\partial \mathbf{x}} (\mathbf{A}(\mathbf{x}^{(k)}) - \mathbf{y}) \right) \quad (1.6)$$

where the weights in the subnetwork at each iteration, k , can be tied or trained independently. A range of different optimization algorithms have been unrolled in this manner, including primal-dual methods [14] and gradient solvers for variational losses [15]. Such networks tend to perform better than simple back-projection networks.

In each case, for best results, training tends to be performed in an end-to-end manner using (1.3).

1.3.2 Limitations of Supervised Learning

While a supervised learning approach seems to offer the possibility of learning an approximation to the statistically optimal estimator, this is based on access to large quantities of ground truth data on which to train the model. This restricts its application to problems that have essentially been already solved previously (in order to generate the ground truth) and is particularly problematic in important scientific, medical and engineering settings, such as astronomical imaging or microscopy and for systems working in complex environments, where ground truth data is scarce and where prediction accuracy is of overriding importance.

One solution that is often adopted in the machine learning community to counter a lack of ground truth training data is to generate data from simulation. Although this provides access to potentially infinite quantities of data, such data is limited to the model from which the simulations are generated and even advanced simulations cannot fully capture the subtle complexities and dependencies that exist in the real setting.

A related issue is the problem of distribution shift, where there is a change between the distribution of the training data and the measurements acquired at test time. For example, ground truth data may be available for a different but related set of signals or images that do not exactly represent the signals being targeted at test time. Even when ground truth data is apparently available, such data is often generated through extended or repeated acquisitions, e.g., in MRI, or increased levels of illumination/radiation, such as in x-ray imaging or microscopy. This can significantly affect the nature of the imaging process and also result in a distribution shift between the acquired training data and the measurements acquired at test time. Unfortunately, supervised learning is notoriously poor at generalizing to such distribution shift [16].

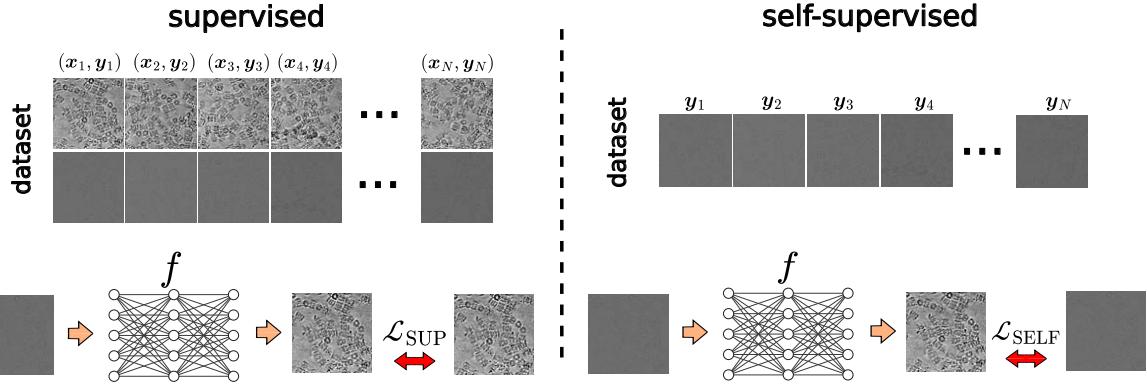


Figure 1.1: **Supervised and self-supervised learning.** Supervised learning requires a dataset of paired data $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, whereas self-supervised learning, the main focus of this manuscript, relies on measurement data alone $\{\mathbf{y}_i\}_{i=1}^N$, and consists of constructing losses that do not require ground truth data, and can approximate the supervised loss.

These challenges have led researchers to seek to develop new self-supervised learning methods that rely solely on the measurement data and knowledge of the acquisition process. Whether trained on just measurements from scratch or used to fine-tune existing models trained on simulations or related data [17], such methods offer the potential in scientific imaging to learn to image structures and patterns for which no ground truth images yet exist [18].

1.4 Self-supervised learning

The essential goal of self-supervised learning methods in imaging and sensing inverse problems is to replace a desired supervised loss function in (1.4), $\mathcal{L}_{\text{SUP}}(\mathbf{x}, \mathbf{y}, f)$, with an self-supervised loss, $\mathcal{L}(\mathbf{y}, f)$, that is only a function of the measurement data, as illustrated in Figure 1.1.

The general strategy is to develop a proxy that can be used in a self-supervised manner to either replicate or approximate the supervised loss. Here, we will see that the acquisition physics and noise model play an essential role in enabling one to formulate such an appropriate self-supervised loss. Examples of the growing body of work include applications to audio restoration [19], point cloud [20] and image denoising [21–24], and image reconstruction [25–27].

In many applications, we can expect to have many $n \gg 1$ samples/pixels, while the dominant statistical dependencies tend to be local. Hence the law of large numbers tells us that even when considering the loss for a single training sample $\mathcal{L}(\mathbf{y}, f) \approx \mathbb{E}_{\mathbf{y}} \{\mathcal{L}(\mathbf{y}, f)\}$ and we can achieve relatively stable estimates of the expected losses with a modest number of samples N (and can sometimes even get away with a single sample). Thus, in most of the analyzes in this monograph, we will assume that we have access to a sufficiently large dataset of measurements $\{\mathbf{y}_i\}_{i=1}^N$ such that we can replace sums over the dataset by expectations over the measurement distribution $p_{\mathbf{y}}$. The effects of having a finite training dataset are discussed in Chapter 4.

Depending on the knowledge of the noise distribution and the range of forward opera-

tors giving rise to the measurements, we can obtain different levels of approximation of the supervised loss.

Unbiased losses The best we can hope for is to build a self-supervised loss that is an unbiased estimate of the supervised loss, i.e.,

$$\mathbb{E}_{\mathbf{y}} \{\mathcal{L}(\mathbf{y}, f)\} = \mathbb{E}_{\mathbf{x}, \mathbf{y}} \{\mathcal{L}_{\text{SUP}}(\mathbf{x}, \mathbf{y}, f)\} + \text{const.} \quad (1.7)$$

where the constant is generally a function of the variance of the noise. Here we can expect to learn a reconstruction function that is as good as the one learned with supervised learning, as long as we have enough measurement data.

In some specific cases, we can have an even stronger result, where the loss is an unbiased estimate of the reconstruction error of a single instance, \mathbf{x} , that is

$$\mathbb{E}_{\mathbf{y}|\mathbf{x}} \{\mathcal{L}(\mathbf{y}, f)\} = \mathbb{E}_{\mathbf{y}|\mathbf{x}} \{\mathcal{L}_{\text{SUP}}(\mathbf{x}, \mathbf{y}, f)\} + \text{const.} \quad (1.8)$$

where the constant is known. This means that we can also use the self-supervised loss to quantify the reconstruction error of \mathbf{x} at test time. This is the case, for example, with Stein's Unbiased Risk Estimator and its variants which will be discussed in Chapter 2.

Constrained losses In some cases, we will not be able to build a loss that is unbiased over the whole space of possible reconstruction functions, but can obtain unbiased estimates over a constrained set of reconstruction functions, \mathcal{F} :

$$\mathbb{E}_{\mathbf{y}} \{\mathcal{L}(\mathbf{y}, f)\} = \mathbb{E}_{\mathbf{x}, \mathbf{y}} \{\mathcal{L}_{\text{SUP}}(\mathbf{x}, \mathbf{y}, f)\} + \text{const.} \text{ for } f \in \mathcal{F} \quad (1.9)$$

The choice of the constraint set can be motivated by either a restricted function class designed to make the problem learnable, e.g., [21, 28], or to incorporate additional prior information within the model, such as imposing an equivariance constraint [26]. If the optimal supervised reconstruction function does not belong to the constrained set \mathcal{F} , the learned reconstruction function will inevitably not be optimal and will not match the performance achievable through supervised learning. However, in some cases we will be able to quantify the bias introduced by the constraint and therefore control the performance gap.

Losses sharing global minimum A final case is when the self-supervised loss is not an unbiased estimate of the supervised loss, but the two losses share a common global minimum, i.e.,

$$\arg \min_f \mathbb{E}_{\mathbf{y}} \{\mathcal{L}(\mathbf{y}, f)\} = \arg \min_f \mathbb{E}_{\mathbf{x}, \mathbf{y}} \{\mathcal{L}_{\text{SUP}}(\mathbf{x}, \mathbf{y}, f)\} \quad (1.10)$$

Thus, we can expect to learn a reconstruction function that is close to the one learned with supervised learning, but may not be able to quantify the associated reconstruction error.

Throughout this survey we will mainly focus on proxies for the ℓ_2 supervised loss (1.4) as this is where the theory is most well developed. However, along the way we will highlight where the theory extends beyond ℓ_2 and/or where practitioners have applied similar techniques using other loss functions in a more heuristic manner.

1.4.1 Learning a generative model

Going beyond self-supervised proxies for the supervised loss, we can ask whether it is possible to learn a generative model for the full signal distribution $p_{\mathbf{x}}(\mathbf{x})$ or the posterior distribution $p(\mathbf{x}|\mathbf{y})$ from measurement data alone. If we were able to learn such a generative model, we would be able to compute not only the conditional mean $f^*(\mathbf{x}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{\mathbf{x}\}$ but also any other posterior statistic. We will see that this is indeed possible in some scenarios, and it often also relies on training on a self-supervised loss that approximates the ℓ_2 supervised loss (e.g., diffusion models require learning a conditional mean estimator). However, it is worth noting that learning a generative model is typically a harder task, and in some cases, learning such a model can be impossible even when constructing a self-supervised loss that approximates the supervised one is still possible [29] (see also Appendix B for some examples). The question of whether it is possible (or not) to identify the signal distribution from measurement data alone is discussed in Section 3.4.

1.5 What this manuscript is not about

Self-supervised representation learning It is important to draw a distinction at this point between the notion of self-supervised learning in imaging and sensing covered in this manuscript, and self-supervised representation learning (SSRL) techniques [30], such as SimCLR [31], BYOL [32], DINO [33], or masked autoencoders [34], that learn powerful representations by training on a set of pretext tasks.

SSRL methods typically require a dataset of clean data $\{\mathbf{x}_i\}_{i=1}^N$, and aim to learn powerful high-level representations for downstream tasks such as classification or segmentation. On the contrary, the self-supervised methods presented here rely on noisy and/or incomplete data alone $\{\mathbf{y}_i\}_{i=1}^N$ and aim to recover the underlying clean images associated to these measurements. Moreover, self-supervised losses in this manuscript serve as proxies for the gold standard supervised reconstruction loss in (1.3), whereas pretext tasks used in SSRL do not aim at approximating a supervised classification or segmentation loss.

Despite these differences, some of the fundamental principles behind the design of pretext tasks, such as invariance to transformations or masking, are also pillars of the self-supervised losses used for imaging inverse problems, and a better understanding of the connections between these two fields remains an open research problem.

Deep image prior and inductive bias While we will generally focus on the behaviour of the expected loss, in practice, there will only be a finite amount of training data and thus the inductive bias of the learning system will also play an important role on actual observed performance.

There are various sources of inductive bias in neural network systems, from the choice of model architecture and weight initialization, to the inclusion of regularization terms in the loss function, e.g., weight decay, and even the optimization procedure, e.g., Adam versus stochastic gradient descent, or the use of early stopping.

For example, Ulyanov et al. [35] showed that various convolutional neural network architectures could be trained to solve inverse problems from a single set of measurements (the one being restored). Something they called the deep image prior (DIP). This has motivated

many researchers [36–38] to try to exploit this concept for unsupervised image reconstruction. However, the nature of the inductive bias is poorly understood [39], and while, as demonstrated in the original DIP paper, the performance is highly dependent on the specific network architecture and is generally well below that obtained by self-supervised methods covered in this manuscript, e.g., see practical comparisons in [40]. Thus, although the DIP is certainly an intriguing phenomenon, we do not consider it further here.

Pretrained diffusion and plug-and-play models Denoising diffusion models [41] and plug-and-play (PnP) solutions [42] have become popular for solving inverse problems. Such methods typically rely on *pre-trained* denoising neural networks, where the denoisers are used to define an implicit signal prior through the score function and Tweedie’s formula. While these solutions are often termed unsupervised, this is not wholly accurate as the creation of the pre-trained denoisers requires access to ground truth data. Nonetheless, there have been recent efforts to learn the denoiser in a self-supervised way [43, 44], which we will also cover in this manuscript.

1.6 Outline

Section 1.6.1 sets out the notation that is commonly used throughout the survey. The outline of the rest of the survey is set out below. Most of the self-supervised methods discussed in this monograph are implemented in the DeepInverse open-source library [45], which contains various jupyter notebook examples, covering many of the topics in this manuscript.

Chapter 2 focuses on the problem of self-supervised learning for denoising with the forward operator, $\mathbf{A} = \mathbf{I}$. We consider various self-supervised losses that act as proxies for the supervised loss under a range of different noise models, from presumed knowledge of various well known noise distributions (Gaussian, Poisson, etc.) to partially specified noise models. Throughout, links between newly proposed self-supervised learning strategies and theoretical results from classical statistics are highlighted. We end by considering the case of more general but invertible forward operators.

Chapter 3 goes on to consider what can be done when we have incomplete measurements, i.e., the forward operator is not invertible. Here, we focus on the case of linear forward operators where there exists a non-trivial null space. We describe two approaches to solving this problem. The first relies on access to a set of multiple forward operators $\{\mathbf{A}_g\}_{g=1}^G$, such as the case of being able to select different sampling patterns in accelerated MRI [27], where the different operators typically have distinct nullspaces. The second approach tackles the more challenging problem of a single rank-deficient forward operator, \mathbf{A} , and instead leverages the assumption that the distribution of signals of interest is invariant to a group of transformations, e.g., a shifted version of an image is still a viable image.

Chapter 4 While the previous two chapters concentrate of the roles of the expected loss functions in enabling self-supervised learning solutions, this chapter considers how accurately these expectations can be approximated when there is only access to finite number of training samples. We consider the simple Noise2Noise algorithm to explore how sample complexity

for self-supervised learning behaves in relation to the supervised learning case. We also show how the standard holdout method used in most supervised learning to avoid overfitting can be extended to the self-supervised learning setting and how pretrained models can be used to reduce the number of measurement samples required for good performance.

Chapter 5 sets out some open problems within the field and possible future research directions.

1.6.1 Notation

Following standard mathematical notation, vectors will be represented by bold lowercase letters and matrices will be represented in bold uppercase. The i th component of a vector \mathbf{x} is written as x_i . The identity matrix is written as \mathbf{I} , the transpose of a matrix, \mathbf{A} , is denoted by \mathbf{A}^\top and its pseudo-inverse is written as \mathbf{A}^\dagger . Other notation that is regularly used throughout the manuscript can be found in the table below.

Symbol	Description
\mathbb{R}	Set of real numbers.
\mathbb{C}	Set of complex numbers.
$p_{\mathbf{x}}(\mathbf{x})$	Signal distribution.
$p_{\mathbf{y}}(\mathbf{y})$	Measurement distribution.
\mathcal{X}	Support of the signal distribution.
\mathcal{Y}	Support of the measurement distribution.
\mathbf{x}	Vector representing the ground truth signal or image.
\mathbf{y}	Vector representing the observed measurements.
\mathbf{b}	Binary vector representing a mask.
n	Dimension of the signal vector, $\mathbf{x} \in \mathbb{R}^n$.
m	Dimension of the measurement vector, $\mathbf{y} \in \mathbb{R}^m$.
k	Dimension of the signal set, \mathcal{X} .
N	Number of training samples.
\mathbf{A}	Forward operator, $\mathbf{A} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ where $\mathbf{y} = \mathbf{A}(\mathbf{x})$.
f	Reconstruction network mapping \mathbf{y} to an estimate of \mathbf{x} .
θ	Weights of a neural network, f .
$\ \cdot\ $	ℓ_2 norm.
$\ \cdot\ _F$	Frobenius norm of a matrix.
$\mathbb{E}_{\mathbf{u}}\{g(\mathbf{u})\}$	Expectation of $g(\mathbf{u})$ under the distribution $p(\mathbf{u})$.
$\mathbb{E}_{\mathbf{u} \mathbf{v}}\{g(\mathbf{u}, \mathbf{v})\}$	Expectation of $g(\mathbf{u}, \mathbf{v})$ under the distribution $p(\mathbf{u} \mathbf{v})$.
$\mathbb{V}_{\mathbf{u} \mathbf{v}}\{\mathbf{u}\}$	Variance of \mathbf{u} under the distribution $p(\mathbf{u} \mathbf{v})$.
$\mathcal{L}_{\text{SUP}}(\mathbf{x}, \mathbf{y}, f)$	Supervised loss.
$\mathcal{L}_X(\mathbf{y}, f)$	Self-supervised loss associated with technique X.
∇	Gradient of a scalar field.
const.	Constant term that is not further quantified.
$\mathcal{N}(\mathbf{x}, \Sigma)$	Multivariate Gaussian with mean \mathbf{x} and covariance Σ .
$\mathcal{P}(\mathbf{x})$	Poisson distribution with rate \mathbf{x} .
$\text{Ber}(\mathbf{x})$	Bernouilli distribution with probability $\mathbf{x} \in [0, 1]^n$.

Chapter 2

Learning from noisy measurements

We start by focusing on denoising problems, where the forward operator is simply the identity mapping $\mathbf{A}(\mathbf{x}) = \mathbf{x}$, and thus both images and measurements lie in the same space. We present various self-supervised losses that only require measurement data and aim at approximating the supervised loss in expectation. We show that the design of the loss is dependent on the knowledge about the noise distribution: if we fully know the noise distribution, we are generally able to build unbiased estimators of the supervised loss, whereas when the noise distribution is not fully known, we can still build self-supervised losses, but they do not achieve the same performance as supervised learning.

The chapter is divided in three parts: in the first part we assume that we observe two independent noisy realizations $(\mathbf{y}_1, \mathbf{y}_2)$ per image \mathbf{x} . In the second part, we will relax this assumption, only relying on a single noisy realization \mathbf{y} per image, but instead assume full knowledge about the noise distribution. In the third part, we will tackle the case where we observe a single noisy realization \mathbf{y} per image and the noise distribution is partially unknown. Most of the results presented here are independent of the architecture or parameterization of the reconstruction network f .

2.1 Learning from independent noisy pairs

In some applications, it is possible to observe two (or more) independent noisy realizations $(\mathbf{y}_1, \mathbf{y}_2)$ of the same underlying signal \mathbf{x} . These can then be used to learn an estimator in a self-supervised way even without explicit knowledge of the noise distribution [46]. Noise2Noise [40] proposed such an approach¹ using one of the noisy measurements as input to the reconstruction network, and the other as target, building the following loss:

$$\mathcal{L}_{\text{N2N}}(\mathbf{y}_1, \mathbf{y}_2, f) = \frac{1}{n} \|f(\mathbf{y}_1) - \mathbf{y}_2\|^2. \quad (\text{Noise2Noise})$$

Since the input noise is independent of the output noise, we can show that (Noise2Noise) is an unbiased estimator of the supervised loss up to a constant:

¹The idea of using independent observations of the same underlying parameter for model selection can be traced back to Mallows work in the 1970s [46]. This idea has been rediscovered in the computer vision field by Noise2Noise [40].

Proposition 2.1. Let \mathbf{y}_1 and \mathbf{y}_2 be two random variables independent conditional on \mathbf{x} , and assume that $\mathbb{E}_{\mathbf{y}_2|\mathbf{x}} \{\mathbf{y}_2\} = \mathbf{x}$, then

$$\mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2|\mathbf{x}} \left\{ \frac{1}{n} \|f(\mathbf{y}_1) - \mathbf{y}_2\|^2 \right\} = \mathbb{E}_{\mathbf{y}_1|\mathbf{x}} \left\{ \frac{1}{n} \|f(\mathbf{y}_1) - \mathbf{x}\|^2 \right\} + \text{const.} \quad (2.1)$$

where the constant is independent of f .

Proof.

$$\begin{aligned} & \mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2|\mathbf{x}} \{\|f(\mathbf{y}_1) - \mathbf{y}_2\|^2\} \\ &= \mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2|\mathbf{x}} \{\|(f(\mathbf{y}_1) - \mathbf{x}) - (\mathbf{y}_2 - \mathbf{x})\|^2\} \\ &= \mathbb{E}_{\mathbf{y}_1|\mathbf{x}} \{\|f(\mathbf{y}_1) - \mathbf{x}\|^2\} - 2 \mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2|\mathbf{x}} \{(f(\mathbf{y}_1) - \mathbf{x})^\top (\mathbf{y}_2 - \mathbf{x})\} + \text{const.} \\ &= \mathbb{E}_{\mathbf{y}_1|\mathbf{x}} \{\|f(\mathbf{y}_1) - \mathbf{x}\|^2\} - 2 (\mathbb{E}_{\mathbf{y}_1|\mathbf{x}} \{f(\mathbf{y}_1) - \mathbf{x}\})^\top (\mathbb{E}_{\mathbf{y}_2|\mathbf{x}} \{\mathbf{y}_2 - \mathbf{x}\}) + \text{const.} \\ &= \mathbb{E}_{\mathbf{y}_1|\mathbf{x}} \{\|f(\mathbf{y}_1) - \mathbf{x}\|^2\} + \text{const.} \end{aligned}$$

where the fourth line uses the fact that \mathbf{y}_1 and \mathbf{y}_2 are conditionally independent and the last line relies uses $\mathbb{E}_{\mathbf{y}_2|\mathbf{x}} \{\mathbf{y}_2 - \mathbf{x}\} = \mathbf{0}$. \square

This result can be extended to any Bregman divergence beyond the ℓ_2 norm [47], but it does not hold for some other popular losses such as the ℓ_1 norm. Intuitively, the estimator f cannot overfit the noise in \mathbf{y}_2 as it observes an independent noise realization \mathbf{y}_1 . The result in Proposition 2.1 requires minimal assumptions on the noise (only that the target has zero-mean noise, i.e., $\mathbb{E}_{\mathbf{y}_2|\mathbf{x}} \{\mathbf{y}_2\} = \mathbf{x}$), making it very appealing for real-world problems where the noise distribution is not known and is possible to obtain two independent observations of the same object. We illustrate this with some imaging examples:

- In cryo-electron microscopy, we observe a series of very noisy images (micrographs) of the same underlying object. Bepler et al. [48] show that we can drastically boost the SNR using a Noise2Noise approach.
- In synthetic aperture radar (SAR), we observe complex images following a circularly-symmetric complex normal distribution, where the real and imaginary parts have independent noise. Dalsasso et al. [49] show that it is possible to train a denoiser with real part as input and imaginary as target.
- In video denoising, the similarity between consecutive frames almost meets the Noise2Noise criterion. Ehret et al. [50] show that a pretrained video denoising network can be fine-tuned using the Noise2Noise approach in combination with optical flow estimates to warp one frame onto another.

The assumption of observing two independent measurements is not met in many applications. Nonetheless, we will see in the following section that (perhaps surprisingly!), if the noise distribution is known, we can often obtain two independent noise realizations $(\mathbf{y}_1, \mathbf{y}_2)$ from a single measurement \mathbf{y} without knowledge of the underlying image \mathbf{x} , and apply the same Noise2Noise loss using these independent pairs.

2.2 Known noise distribution

In many applications, the noise distribution is approximately known, or it can be approximated using some calibration data. There are two main approaches for building self-supervised losses that incorporate this knowledge: the first approach was pioneered by Noisier2Noise [51] and Recorrupted2Recorrupted [52], who showed that it is possible to add synthetic noise to the observation \mathbf{y} to generate two independent realizations $(\mathbf{y}_1, \mathbf{y}_2)$. A second approach is based on a classical result in statistics known as Stein's Unbiased Risk Estimate (SURE) [53], which penalizes the divergence of the network f to avoid overfitting the noise. In both cases, we require exact knowledge of the noise distribution in order to correctly approximate the supervised case. We will further see that, despite at first sight looking quite different, Recorrupted2Recorrupted and SURE are closely related.

2.2.1 Bootstrapping noisy measurements

While the Noisier2Noise framework [51] set out the original approach to bootstrapping noisy measurements, we will follow the equivalent² Recorrupted2Recorrupted [52] work as this sets the scene for further generalizations.

Assuming a Gaussian noise model, $\mathbf{y} = \mathbf{x} + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, or equivalently that $\mathbf{y} \sim \mathcal{N}(\mathbf{x}, \boldsymbol{\Sigma})$, we can resample two independent noisy realizations from the original measurement, \mathbf{y} , as

$$\begin{cases} \mathbf{y}_1 = \mathbf{y} + \sqrt{\frac{\alpha}{1-\alpha}} \boldsymbol{\omega} \\ \mathbf{y}_2 = \mathbf{y} - \sqrt{\frac{1-\alpha}{\alpha}} \boldsymbol{\omega} \end{cases} \quad (2.2)$$

where $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ follows the same distribution as the noise $\boldsymbol{\epsilon}$ and $\alpha \in (0, 1)$ is a positive scalar parameter.

Proposition 2.2 (Pang et al. [52]). *The random variables \mathbf{y}_1 and \mathbf{y}_2 defined by (2.2) are independent conditional on \mathbf{x} for any $\alpha \in (0, 1)$.*

Proof. Let $\tau = \sqrt{\frac{\alpha}{1-\alpha}}$. Since \mathbf{y}_1 and \mathbf{y}_2 follow a Gaussian distribution conditional on \mathbf{x} , we can prove their independence by simply showing that they are not linearly correlated:

$$\begin{aligned} & \mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2 | \mathbf{x}} \left\{ (\mathbf{y}_1 - \mathbf{x})(\mathbf{y}_2 - \mathbf{x})^\top \right\} \\ &= \mathbb{E}_{\boldsymbol{\epsilon}, \boldsymbol{\omega}} \left\{ (\boldsymbol{\epsilon} + \tau \boldsymbol{\omega})(\boldsymbol{\epsilon} - \frac{1}{\tau} \boldsymbol{\omega})^\top \right\} \\ &= \mathbb{E}_{\boldsymbol{\epsilon}} \left\{ \boldsymbol{\epsilon} \boldsymbol{\epsilon}^\top \right\} - \frac{1}{\tau} \mathbb{E}_{\boldsymbol{\epsilon}} \left\{ \boldsymbol{\epsilon} \right\} \mathbb{E}_{\boldsymbol{\omega}} \left\{ \boldsymbol{\omega}^\top \right\} + \tau \mathbb{E}_{\boldsymbol{\omega}} \left\{ \boldsymbol{\omega} \right\} \mathbb{E}_{\boldsymbol{\epsilon}} \left\{ \boldsymbol{\epsilon}^\top \right\} - \mathbb{E}_{\boldsymbol{\omega}} \left\{ \boldsymbol{\omega} \boldsymbol{\omega}^\top \right\} \\ &= \mathbb{E}_{\boldsymbol{\epsilon}} \left\{ \boldsymbol{\epsilon} \boldsymbol{\epsilon}^\top \right\} - \mathbb{E}_{\boldsymbol{\omega}} \left\{ \boldsymbol{\omega} \boldsymbol{\omega}^\top \right\} \\ &= \mathbf{0} \end{aligned}$$

The last line relies on the assumption that the added noise $\boldsymbol{\omega}$ has the same covariance as the measurement noise to achieve independence. \square

²Noisier2Noise [51] introduced the idea of adding noise to the inputs previous to Recorrupted2Recorrupted [52], but the latter presented a simplified loss, showing conditional independence of the simulated pairs $(\mathbf{y}_1, \mathbf{y}_2)$. See Appendix A for more details regarding the close links between these approaches.

Following the Noise2Noise approach, we can define the Recorrupted2Recorrupted loss as

$$\mathcal{L}_{\text{R2R}}(\mathbf{y}, f) = \mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2 | \mathbf{y}} \left\{ \frac{1}{n} \|f(\mathbf{y}_1) - \mathbf{y}_2\|^2 \right\} \quad (\text{R2R})$$

which, due to the conditional independence of $(\mathbf{y}_1, \mathbf{y}_2)$ and $\mathbb{E}_{\mathbf{y}_2 | \mathbf{x}} \{\mathbf{y}_2\} = \mathbf{x}$, is an unbiased estimate of the supervised ℓ_2 loss with \mathbf{y}_1 at the input of the network:

$$\mathbb{E}_{\mathbf{y} | \mathbf{x}} \{\mathcal{L}_{\text{R2R}}(\mathbf{y}, f)\} = \mathbb{E}_{\mathbf{y}_1 | \mathbf{x}} \left\{ \frac{1}{n} \|f(\mathbf{y}_1) - \mathbf{x}\|^2 \right\} + \text{const.} \quad (2.3)$$

Note that (R2R) is an idealized loss, as it involves the expectation over the resampled realizations. However, in practice we can use a single resampled pair, $(\mathbf{y}_1, \mathbf{y}_2)$, per gradient step and the resulting stochastic gradient estimates of the loss will remain unbiased.

The independence of \mathbf{y}_1 and \mathbf{y}_2 also comes at a price: the input to the network, \mathbf{y}_1 , has lower signal-to-noise ratio (SNR) than the original measurement, \mathbf{y} , due to the additional synthetic noise. The parameter $\alpha \in (0, 1)$ acts as a trade-off between the amount of additional noise injected into \mathbf{y}_1 and \mathbf{y}_2 . By defining the SNR of a random variable \mathbf{z} as $\text{SNR}(\mathbf{z}) := \frac{\|\mathbb{E}_{\mathbf{z}}\{\mathbf{z}\}\|^2}{\mathbb{E}_{\mathbf{z}}\{\|\mathbf{z} - \mathbb{E}_{\mathbf{z}}\{\mathbf{z}\}\|^2\}}$ we have that the SNR of the new variables is

$$\begin{aligned} \text{SNR}(\mathbf{y}_1) &= (1 - \alpha) \text{SNR}(\mathbf{y}), \\ \text{SNR}(\mathbf{y}_2) &= \alpha \text{SNR}(\mathbf{y}). \end{aligned} \quad (2.4)$$

Thus, a good strategy is to choose α small, to have as much SNR on the input \mathbf{y}_1 as possible, but not too small, to avoid targets \mathbf{y}_2 with very low SNR, which would result in noisier loss estimates. In practice we only use a finite number of resamplings of pairs $(\mathbf{y}_1, \mathbf{y}_2)$ for a fixed \mathbf{y} . A good choice in practice generally lies in the interval $\alpha \in (0.05, 0.2)$. For more information about the optimal choice of α , see the analysis by Oliveira et al. [54].

At test time, we can improve the estimation by averaging over $J > 1$ additions of synthetic noise, that is

$$f^{\text{test}}(\mathbf{y}) = \frac{1}{J} \sum_{j=1}^J f(\mathbf{y}_1^{(j)}) \quad (2.5)$$

where $\mathbf{y}_1^{(j)} \sim \mathcal{N} \left(\mathbf{y}, \frac{\alpha}{1-\alpha} \boldsymbol{\Sigma} \right)$ for $j = 1, \dots, J$.

This loss can be extended to non-Gaussian additive noise: if some first order moments of the noise distribution are known, we can still use (2.2) to generate pairs $(\mathbf{y}_1, \mathbf{y}_2)$ by adding synthetic noise $\boldsymbol{\omega}$ that matches these moments [55].

Extensions beyond additive noise In many applications, the noise affecting the measurements is not additive. For example, Poisson noise arises in photon-counting devices such as single-photon lidar [56], and the Gamma distribution is often used to model speckle noise associated with synthetic aperture radar images [49].

Gaussian, Poisson and Gamma distributions belong to a natural exponential family (NEF) [57], and can be written as

$$p(\mathbf{y} | \mathbf{x}) = h(\mathbf{y}) \exp(\mathbf{y}^\top \eta(\mathbf{x}) - \phi(\mathbf{x})), \quad (\text{NEF})$$

for some h , η and ϕ functions which are specific to each distribution (note this forms a NEF with respect to the *natural parameter*, $\eta(\mathbf{x})$ and not with respect to the image, \mathbf{x} , itself). We can generalize the (R2R) loss to the NEF using the following theorem:

Theorem 2.3 (Monroy et al. [55]). *Let $p(\mathbf{y}|\mathbf{x})$ belong to the NEF with $\mathbb{E}_{\mathbf{y}|\mathbf{x}}\{\mathbf{y}\} = \mathbf{x}$ and $\alpha \in (0, 1)$. We can sample \mathbf{y}_1 and \mathbf{y}_2 as*

$$\begin{cases} \mathbf{y}_1 \sim p(\mathbf{y}_1|\mathbf{y}, \alpha), \\ \mathbf{y}_2 = \frac{1}{\alpha}\mathbf{y} - \frac{(1-\alpha)}{\alpha}\mathbf{y}_1, \end{cases} \quad (2.6)$$

such that \mathbf{y}_1 and \mathbf{y}_2 are independent random variables conditional on \mathbf{x} , with means $\mathbb{E}_{\mathbf{y}_1|\mathbf{x}}\{\mathbf{y}_1\} = \mathbb{E}_{\mathbf{y}_2|\mathbf{x}}\{\mathbf{y}_2\} = \mathbf{x}$ and variances $\mathbb{V}_{\mathbf{y}_1|\mathbf{x}}\{\mathbf{y}_1\} = \frac{1}{1-\alpha}\mathbb{V}_{\mathbf{y}|\mathbf{x}}\{\mathbf{y}\}$ and $\mathbb{V}_{\mathbf{y}_2|\mathbf{x}}\{\mathbf{y}_2\} = \frac{1}{\alpha}\mathbb{V}_{\mathbf{y}|\mathbf{x}}\{\mathbf{y}\}$, and their distributions $p_1(\mathbf{y}_1|\mathbf{x})$ and $p_2(\mathbf{y}_2|\mathbf{x})$ also belong to the NEF.

The generalized R2R loss is thus defined as

$$\mathcal{L}_{\text{GR2R}}(\mathbf{y}, f) = \mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2|\mathbf{y}} \left\{ \frac{1}{n} \|f(\mathbf{y}_1) - \mathbf{y}_2\|^2 \right\} \quad (\text{GR2R})$$

where \mathbf{y}_1 and \mathbf{y}_2 are generated via (2.6). Since the synthetic pairs are independent conditional on \mathbf{x} , we can use again Proposition 2.1 to show that

$$\mathbb{E}_{\mathbf{y}|\mathbf{x}}\{\mathcal{L}_{\text{GR2R}}(\mathbf{y}, f)\} = \mathbb{E}_{\mathbf{y}_1|\mathbf{x}}\{\|f(\mathbf{y}_1) - \mathbf{x}\|^2\} + \text{const.}$$

The definition³ of $p(\mathbf{y}_1|\mathbf{y}, \alpha)$ for the Gaussian, Poisson and Gamma noise distributions is included in Table 2.1. As in the Gaussian noise case, the SNR of \mathbf{y}_1 and \mathbf{y}_2 is given by (2.4), and α should be chosen to approximately lie in the (0.05, 0.2) interval [55].

The idea of generating synthetic pairs of independent data has also been recently explored in other statistical inference applications where it is known as *data fission* [58] and includes extensions beyond the NEF.

Beyond the ℓ_2 loss We can also incorporate the knowledge about the distribution $p(\mathbf{y}_2|\mathbf{x})$ in the choice of the metric of the loss. The ℓ_2 metric is well adapted for Gaussian loss, since it is proportional to the negative log-likelihood under the Gaussian model. Thus, if the noise model is not Gaussian but belongs to the NEF, we can use the negative log-likelihood of the appropriate noise model:

$$\begin{aligned} \mathcal{L}_{\text{GR2R-NLL}}(\mathbf{y}, f) &= \mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2|\mathbf{y}} \left\{ \frac{1}{n} \phi(f(\mathbf{y}_1)) - \frac{1}{n} \mathbf{y}_2^\top \eta(f(\mathbf{y}_1)) \right\} \\ &= \mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2|\mathbf{y}} \left\{ -\frac{1}{n} \log p_2(\mathbf{y}_2|\hat{\mathbf{x}} = f(\mathbf{y}_1)) \right\} + \text{const.} \end{aligned}$$

³In general, we have that $p(\mathbf{y}_1|\mathbf{y}, \alpha) = h_1(\mathbf{y}_1)h_2(\mathbf{y} - \mathbf{y}_1)/h(\mathbf{y})$ where

$$\begin{aligned} h_1(\mathbf{y}_1) &= \int e^{-\mathbf{s}^\top \mathbf{y}_1 + (1-\alpha)\phi(\eta^{-1}(\frac{\mathbf{s}}{1-\alpha}))} d\mathbf{s} \\ h_2(\mathbf{y}_2) &= \int e^{-\mathbf{s}^\top \mathbf{y}_2 + \alpha\phi(\eta^{-1}(\frac{\mathbf{s}}{\alpha}))} d\mathbf{s}. \end{aligned}$$

Model	Gaussian $\mathbf{y} \sim \mathcal{N}(\mathbf{x}, \Sigma)$	Poisson $z \sim \mathcal{P}(\mathbf{x}/\gamma), \mathbf{y} = \gamma z$	Gamma $\mathbf{y} \sim \mathcal{G}(\ell, \ell/\mathbf{x})$
\mathbf{y}_1	$\mathbf{y}_1 = \mathbf{y} + \sqrt{\frac{\alpha}{1-\alpha}} \boldsymbol{\omega},$ $\boldsymbol{\omega} \sim \mathcal{N}(0, \Sigma)$	$\mathbf{y}_1 = \frac{\mathbf{y} - \gamma \boldsymbol{\omega}}{1-\alpha},$ $\boldsymbol{\omega} \sim \text{Bin}(z, \alpha)$	$\mathbf{y}_1 = \mathbf{y} \circ (1 - \boldsymbol{\omega})/(1 - \alpha)$ $\boldsymbol{\omega} \sim \text{Beta}(\ell\alpha, \ell(1 - \alpha))$
$\mathcal{L}_{\text{GR2R-NLL}}(\mathbf{y}, f)$	$\ \sqrt{\Sigma^{-1}}(f(\mathbf{y}_1) - \mathbf{y}_2)\ ^2$	$-\gamma \mathbf{y}_2^\top \log f(\mathbf{y}_1) + \mathbf{1}^\top f(\mathbf{y}_1)$	$\log f(\mathbf{y}_1) + \mathbf{y}_2/f(\mathbf{y}_1)$
$\mathcal{L}_{\text{SURE}}(\mathbf{y}, f) = \lim_{\alpha \rightarrow 0} \mathcal{L}_{\text{GR2R}}(\mathbf{y}, \alpha, f)$	$\ f(\mathbf{y}) - \mathbf{y}\ ^2 + 2 \text{trace}(\Sigma \frac{\partial f}{\partial \mathbf{y}}(\mathbf{y}))$	$\ f(\mathbf{y}) - \mathbf{y}\ ^2 + 2 \sum_{i=1}^n y_i(f_i(\mathbf{y}) - f_i(\mathbf{y} - \gamma \mathbf{e}_i))$	$\ f(\mathbf{y}) - \mathbf{y}\ ^2 + 2 \sum_{i=1}^n \sum_{k \geq 1} b(\ell, k)(-y_i)^{k+1} \frac{\partial^k f_i}{\partial y_i^k}(\mathbf{y})$

Table 2.1: **Summary of generalized R2R losses.** Popular noise distributions belonging to the natural exponential family and the associated bootstrapping functions with $\alpha \in (0, 1)$ and negative-log likelihood losses.

Due to the independence of \mathbf{y}_1 and \mathbf{y}_2 conditional on \mathbf{x} , the loss is an unbiased estimator of a supervised negative log-likelihood loss:

$$\mathbb{E}_{\mathbf{y}|\mathbf{x}} \{ \mathcal{L}_{\text{GR2R-NLL}}(\mathbf{y}, f) \} = \mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ -\frac{1}{n} \log p_2(\mathbf{x}|\hat{\mathbf{x}} = f(\mathbf{y}_1)) \right\} + \text{const.}$$

These losses (including ℓ_2) correspond to Bregman divergences, and share the same global minima in expectation over the dataset, i.e., the posterior mean $f(\mathbf{y}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ \mathbf{x} \}$ [55]. However, when dealing with finite datasets, using the ℓ_2 or the negative log-likelihood will lead to different networks f . The resulting negative log-likelihood losses of (anisotropic) Gaussian, Gamma and Poisson noise distributions are included in Table 2.1.

2.2.2 Stein's Unbiased Risk Estimate

We now turn to a self-supervised loss that is not based on generating two independent noisy pairs. Let us return again to the Gaussian noise model, $\mathbf{y}|\mathbf{x} \sim \mathcal{N}(\mathbf{x}, \Sigma)$. The following seminal result by Stein [53] shows that we can estimate the correlation between the prediction and the noise without knowledge of the ground truth:

Lemma 2.4 (Stein [53]). *Let $\mathbf{y}|\mathbf{x} \sim \mathcal{N}(\mathbf{x}, \Sigma)$ and f be weakly differentiable. Then, we have*

$$\mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ f(\mathbf{y})^\top (\mathbf{y} - \mathbf{x}) \right\} = \mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ \text{trace} \left(\frac{\partial f}{\partial \mathbf{y}} \Sigma \right) \right\} \quad (2.7)$$

where $\frac{\partial f}{\partial \mathbf{y}} \in \mathbb{R}^{n \times n}$ is the Jacobian of f at \mathbf{y} .

The proof relies on integration by parts, and we leave it to the reader as an exercise. When the noise is isotropic, i.e., $\Sigma = \sigma^2 \mathbf{I}$ with $\text{trace} \left(\frac{\partial f}{\partial \mathbf{y}} \right) = \sum_{i=1}^n \frac{\partial f_i}{\partial y_i}$, this shows that the correlation between the noise and prediction is simply proportional to the divergence of the estimator, f .

Building on this lemma, the Stein's Unbiased Risk Estimate (SURE) is given by:

$$\mathcal{L}_{\text{SURE}}(\mathbf{y}, f) = \frac{1}{n} \|f(\mathbf{y}) - \mathbf{y}\|^2 + \frac{2}{n} \text{trace} \left(\frac{\partial f}{\partial \mathbf{y}} \Sigma \right) - \frac{1}{n} \text{trace}(\Sigma) \quad (\text{SURE})$$

and is an unbiased estimate of the supervised loss as

$$\begin{aligned}
& \mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ \frac{1}{n} \|f(\mathbf{y}) - \mathbf{y}\|^2 + \frac{2}{n} \text{trace} \left(\frac{\partial f}{\partial \mathbf{y}} \boldsymbol{\Sigma} \right) - \frac{1}{n} \text{trace} (\boldsymbol{\Sigma}) \right\} \\
&= \mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ \frac{1}{n} \| (f(\mathbf{y}) - \mathbf{x}) - (\mathbf{y} - \mathbf{x}) \|^2 + \frac{2}{n} \text{trace} \left(\frac{\partial f}{\partial \mathbf{y}} \boldsymbol{\Sigma} \right) - \frac{1}{n} \text{trace} (\boldsymbol{\Sigma}) \right\} \\
&= \mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ \frac{1}{n} \|f(\mathbf{y}) - \mathbf{x}\|^2 - \frac{2}{n} f(\mathbf{y})^\top (\mathbf{y} - \mathbf{x}) + \frac{2}{n} \text{trace} \left(\frac{\partial f}{\partial \mathbf{y}} \boldsymbol{\Sigma} \right) \right\} \\
&= \mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ \frac{1}{n} \|f(\mathbf{y}) - \mathbf{x}\|^2 \right\}
\end{aligned}$$

where the third line uses Stein's lemma and $\mathbb{E}_{\mathbf{y}|\mathbf{x}} \{ \| \mathbf{y} - \mathbf{x} \|^2 \} = \text{trace} (\boldsymbol{\Sigma})$. SURE has been widely popular in the signal processing community well before the advent of neural networks, see e.g., [59]. To the best of our knowledge, Metzler et al. [60] were the first to use SURE for training deep networks, and it has been widely used for learning ever since, e.g., [61, 62].

Extensions beyond Gaussian noise The SURE loss has been extended beyond Gaussian noise. In the case of measurements corrupted by Poisson noise, $\mathbf{y} \sim \gamma \mathcal{P}(\frac{\mathbf{x}}{\gamma})$ with gain $\gamma > 0$, we can use the following extension of Stein's lemma, introduced by Hudson [63]:

$$\mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ (\mathbf{y} - \mathbf{x})^\top f(\mathbf{y}) \right\} = \mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ \sum_{i=1}^n y_i (f_i(\mathbf{y}) - f_i(\mathbf{y} - \gamma \mathbf{e}_i)) \right\} \quad (2.8)$$

where \mathbf{e}_i is a canonical vector containing a one in the i th entry and zeros in the rest. In a similar fashion to (SURE), this lemma can be used to derive the following self-supervised loss [5]:

$$\mathcal{L}_{\text{PURE}}(\mathbf{y}, f) = \frac{1}{n} \|f(\mathbf{y}) - \mathbf{y}\|^2 + \frac{2}{n} \sum_{i=1}^n y_i (f_i(\mathbf{y}) - f_i(\mathbf{y} + \gamma \mathbf{e}_i)) \quad (\text{PURE})$$

which is an unbiased estimator of the supervised loss when the measurements are corrupted by Poisson noise. Le Montagner et al. [64] combined the Stein (2.7) and Hudson (2.8) identities to handle mixed Poisson-Gaussian noise, i.e. $\mathbf{y} \sim \gamma \mathcal{P}(\frac{\mathbf{x}}{\gamma}) + \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, which appears in various imaging applications, such as fluorescence microscopy [65] or low-dose computed tomography [66].

Other similar extensions of SURE to continuous variables belonging to the NEF can be obtained using the following lemma:

Lemma 2.5 (Hudson [63]). *Let $\mathbf{y} \sim p(\mathbf{y}|\mathbf{x})$ be a continuous random variable whose distribution belongs to the (NEF), and let f be weakly differentiable. Then*

$$\mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ (\nabla \log h(\mathbf{y}) + \eta(\mathbf{x}))^\top f(\mathbf{y}) \right\} = \sum_{i=1}^n \mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ \frac{\partial f_i}{\partial y_i}(\mathbf{y}) \right\}. \quad (2.9)$$

Stein's lemma is recovered for the special case of (isotropic) Gaussian noise with $\eta(\mathbf{x}) = -\frac{\mathbf{x}}{\sigma^2}$ and $h(\mathbf{y}) = \exp(\frac{\|\mathbf{y}\|^2}{2\sigma^2})$.

Eldar [67] used this result to build the generalized SURE (GSURE) loss. In the notation of (NEF), the loss can be written as:

$$\mathcal{L}_{\text{GSURE}}(\mathbf{y}, f) = \frac{1}{n} \|f(\mathbf{y}) + \nabla \log h(\mathbf{y})\|^2 + \frac{2}{n} \sum_{i=1}^n \frac{\partial f_i}{\partial y_i}(\mathbf{y}) \quad (\text{GSURE})$$

which, similar to the Gaussian case, we can use Lemma 2.5 to show that

$$\mathbb{E}_{\mathbf{y}|\mathbf{x}} \{\mathcal{L}_{\text{GSURE}}(\mathbf{y}, f)\} = \mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ \frac{1}{n} \|f(\mathbf{y}) - \eta(\mathbf{x})\|^2 \right\} + \text{const.} \quad (2.10)$$

Note, however, this is not suitable for deriving a loss equivalent to those presented in this chapter, as GSURE targets the natural parameter, $\eta(\mathbf{x})$, instead of \mathbf{x} . The optimal f is thus the estimator $f(\mathbf{y}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{\eta(\mathbf{x})\}$ which is not equal to the posterior mean, except for the Gaussian noise case where $\eta(\mathbf{x}) \propto \mathbf{x}$.

Approximating the divergence term For complex models such as neural networks, there are generally no analytic solutions for the divergence term in the SURE loss and it is common practice to resort to Monte Carlo estimates. A first option is to compute the divergence term in the SURE loss using automatic differentiation [68], together with Hutchinson's unbiased trace estimator [69]:

$$\text{trace} \left(\Sigma \frac{\partial f}{\partial \mathbf{y}} \right) \approx \boldsymbol{\omega}^\top \frac{\partial f}{\partial \mathbf{y}} \boldsymbol{\omega} \quad (2.11)$$

where $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, \Sigma)$ is a random standard Gaussian, and $\frac{\partial f}{\partial \mathbf{y}} \boldsymbol{\omega}$ is computed using as a Jacobian-vector product via automatic differentiation.

Alternatively, one can use a finite difference approximation to the same estimator via a simple Monte Carlo approximation [70]

$$\text{trace} \left(\Sigma \frac{\partial f}{\partial \mathbf{y}} \right) \approx (\Sigma \frac{\boldsymbol{\omega}}{\tau})^\top (f(\mathbf{y} + \tau \boldsymbol{\omega}) - f(\mathbf{y})) \quad (2.12)$$

where $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and small $\tau > 0$. The approximation becomes exact if we take the expectation over $\boldsymbol{\omega}$ and let $\tau \rightarrow 0$. This approximation avoids the need for automatic differentiation but at the cost of requiring two evaluations of f per calculation. In practice, τ can be chosen to be a small fraction, for example 1%, of the standard deviation of the noise.

Although unbiased, both these estimators can potentially have high variance which can be reduced by averaging over multiple samples of $\boldsymbol{\omega}$. However, for imaging applications, as argued in [70], it is usually reasonable to assume that a single Monte Carlo sample will provide a sufficiently low variance estimate. Intuitively, this is because denoising functions tend to act locally and we are therefore spatially averaging over a large number, $n \ll 1$ of almost independent estimates per pixel.

In the case of Poisson noise, Luisier et al. [5] proposed the following approximation of the Poisson variant (PURE):

$$\begin{aligned} \sum_{i=1}^n y_i (f_i(\mathbf{y}) - f_i(\mathbf{y} - \gamma e_i)) &\approx \text{trace} \left(\gamma \text{diag}(\mathbf{y}) \frac{\partial f}{\partial \mathbf{y}} \right) \\ &\approx (\gamma \text{diag}(\mathbf{y}) \frac{\boldsymbol{\omega}}{\tau})^\top (f(\mathbf{y} + \tau \boldsymbol{\omega}) - f(\mathbf{y})) \end{aligned} \quad (2.13)$$

The approximation relies on the assumption that the Poisson noise is approximately Gaussian with a signal dependent covariance, that is $\Sigma \approx \text{diag}(\mathbf{y})$ when γ is large, and, it is not well suited for low γ settings.

Equivalence with Recorrupted2Recorrupted The attentive reader might have noticed that the synthetic noise in R2R (2.2) and the Monte Carlo approximation of SURE (2.12) are relatively similar. It turns out that (R2R) can be seen as another Monte Carlo approximation of the analytic (SURE) as $\alpha \rightarrow 0$. This observation was first made by Oliveira et al. [54] for the Gaussian noise case, then extended to the (discrete) Poisson case [71], and finally extended to the continuous natural exponential family in the following proposition:

Proposition 2.6 (Monroy et al. [55]). *Assume that f is analytic, $p(\mathbf{y}|\mathbf{x})$ is a continuous distribution belonging to the NEF, and that $a_k : \mathbb{R} \mapsto \mathbb{R}$ as*

$$a_k(y_i) = \lim_{\alpha \rightarrow 0} \mathbb{E}_{y_{2,i}|y_i,\alpha} \left\{ (y_{2,i} - y_i)(\alpha y_{2,i})^k \right\}$$

for all $i = 1, \dots, n$ verifies $|a_k(y_i)| < \infty$ for all positive integers $k \geq 1$. Then,

$$\begin{aligned} \lim_{\alpha \rightarrow 0} \mathcal{L}_{GR2R}(\mathbf{y}, f, \alpha) &= \\ &\frac{1}{n} \|f(\mathbf{y}) - \mathbf{y}\|^2 + \frac{2}{n} \sum_{i=1}^n \sum_{k \geq 1} (-1)^{k+1} a_k(y_i) \frac{1}{k!} \frac{\partial^k f_i}{\partial y_i^k}(\mathbf{y}) + \text{const.} \end{aligned}$$

where the resulting SURE-like loss is an unbiased estimator of the supervised loss with input \mathbf{y} instead of \mathbf{y}_1 , that is

$$\mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ \lim_{\alpha \rightarrow 0} \mathcal{L}_{GR2R}(\mathbf{y}, f, \alpha) \right\} = \mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ \frac{1}{n} \|f(\mathbf{y}) - \mathbf{x}\|^2 \right\}.$$

Interestingly, in the isotropic Gaussian case we have $a_1(y_i) = \sigma^2$ and $a_k(y_i) = 0$ for $k \geq 2$, recovering the standard SURE formula. In the Poisson noise case [71], the R2R loss recovers (PURE) as $\alpha \rightarrow 0$, without relying on the continuous approximation in (2.13). Unlike (GSURE) that requires a single divergence term but learns the conditional estimator $\mathbb{E}_{\mathbf{x}|\mathbf{y}} \{\eta(\mathbf{x})\}$, Proposition 2.6 shows that SURE-like formulas exist for learning the conditional mean $\mathbb{E}_{\mathbf{x}|\mathbf{y}} \{\mathbf{x}\}$, albeit they often require computing higher-order derivatives of f .

Connection with Tweedie's formula The second term in (SURE) verifies the following equality

$$\mathbb{E}_{\mathbf{y}} \left\{ \text{trace} \left(\frac{\partial f}{\partial \mathbf{y}} \Sigma \right) \right\} = -\mathbb{E}_{\mathbf{y}} \left\{ f(\mathbf{y})^\top \Sigma \nabla \log p_{\mathbf{y}}(\mathbf{y}) \right\}$$

which can be shown again using integration by parts. Using this result, we can find the optimal denoiser under a SURE loss by solving

$$\begin{aligned} f^* &= \arg \min_f \mathbb{E}_{\mathbf{y}} \{ \mathcal{L}_{\text{SURE}}(\mathbf{y}, f) \} \\ &= \arg \min_f \mathbb{E}_{\mathbf{y}} \left\{ \frac{1}{n} \|f(\mathbf{y}) - \mathbf{y}\|^2 - \frac{2}{n} f(\mathbf{y})^\top \Sigma \nabla \log p_{\mathbf{y}}(\mathbf{y}) \right\} \\ &= \arg \min_f \mathbb{E}_{\mathbf{y}} \left\{ \frac{1}{n} \|f(\mathbf{y}) - \mathbf{y} - \Sigma \nabla \log p_{\mathbf{y}}(\mathbf{y})\|^2 \right\} \end{aligned}$$

where the last step completes squares, and $\nabla \log p_{\mathbf{y}}(\mathbf{y})$ is known as *the score* of the measurement distribution. The optimal solution is thus the well-known Tweedie's formula:

$$f^*(\mathbf{y}) = \mathbf{y} + \boldsymbol{\Sigma} \nabla \log p_{\mathbf{y}}(\mathbf{y}) \quad (\text{Tweedie})$$

Since SURE is an unbiased estimator of the supervised ℓ_2 loss, whose global minimizer is the conditional mean estimator $f^*(\mathbf{y}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}}\{\mathbf{x}\}$, we have that the optimal least-squares estimator in the Gaussian noise case is given by $\mathbb{E}_{\mathbf{x}|\mathbf{y}}\{\mathbf{x}\} = \mathbf{y} + \boldsymbol{\Sigma} \nabla \log p_{\mathbf{y}}(\mathbf{y})$.

We can further combine (SURE) and (Tweedie) to compute the minimum mean squared error (MMSE) of the denoising problem, as a function of the score and the noise covariance:

$$\text{MMSE} = \mathbb{E}_{\mathbf{y}}\{\mathcal{L}_{\text{SURE}}(\mathbf{y}, f^*)\} \quad (2.14)$$

$$= \frac{1}{n} \text{trace}(\boldsymbol{\Sigma}) - \mathbb{E}_{\mathbf{y}} \left\{ \frac{1}{n} \|\boldsymbol{\Sigma} \nabla \log p_{\mathbf{y}}(\mathbf{y})\|^2 \right\} \quad (2.15)$$

Inspired by the close link between SURE and Tweedie's formula, the Noise2Score approach [72, 73] proposes to first train a network $s(\mathbf{y})$ that approximates the score, i.e., $s(\mathbf{y}) \approx \nabla \log p_{\mathbf{y}}(\mathbf{y})$, and then applies (Tweedie) at test time $f(\mathbf{y}) = \mathbf{y} + \boldsymbol{\Sigma} s(\mathbf{y})$.

Tweedie's formula also plays a significant role in diffusion models, as it allows one to evaluate the score function indirectly via a denoiser network $f(\mathbf{y})$. Section 2.4 discusses self-supervised diffusion methods that leverage this connection.

2.3 Partially unknown noise distribution

In many real-world settings, we do not have independent pairs $(\mathbf{y}_1, \mathbf{y}_2)$, and thus cannot use (Noise2Noise), nor do we know exactly the noise distribution and thus cannot use (R2R) or (SURE) losses. Under certain circumstances we will see that we can still build self-supervised losses, by paying a price on the flexibility of the learned denoiser: we can only expect to minimize a *constrained* supervised loss [28], that is

$$\arg \min_f \mathbb{E}_{\mathbf{x}, \mathbf{y}} \{ \|f(\mathbf{y}) - \mathbf{x}\|^2 \} \quad \text{s.t. } f \in \mathcal{F} \quad (2.16)$$

where \mathcal{F} is a constrained set of functions. Thus, we generally are not able to approximate the oracle estimator, i.e., the conditional mean $f^*(\mathbf{y}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}}\{\mathbf{x}\}$ if it does not belong to \mathcal{F} . However, as we will see, in some cases the constraints can be relatively mild, letting us find an f performing close to the oracle. As illustrated in Figure 2.1, the less we know about the distribution, the stronger the constraints needed, and we get further away from the supervised performance. For example, we might know the noise is isotropic and Gaussian, but not the noise level σ^2 , or more extreme, we might not know the distribution at all, except for an assumption of independence across pixels.

2.3.1 Unknown noise level Stein's Unbiased Risk Estimate

In some applications, the noise model can be assumed to be Gaussian $\mathbf{y} = \mathbf{x} + \sigma \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ but the noise level σ is unknown. A naive approach could be to estimate σ from the noisy data first, and then train using (SURE) or (R2R). As illustrated in Figure 2.2,

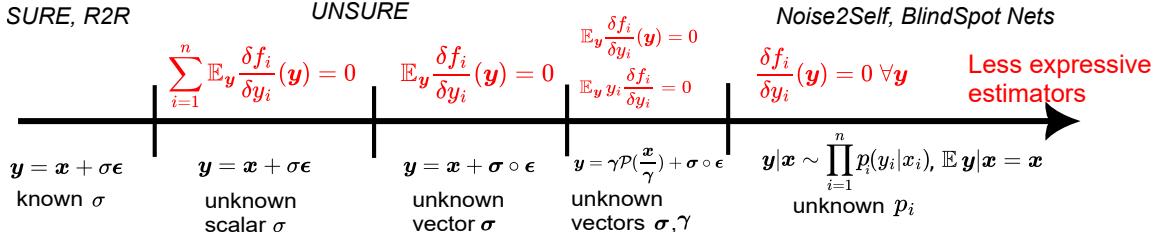


Figure 2.1: **The expressivity-robustness trade-off in self-supervised denoising** [28]. As the assumptions about the noise distribution are relaxed, the learned estimator needs to be less expressive to avoid over-fitting the noise.

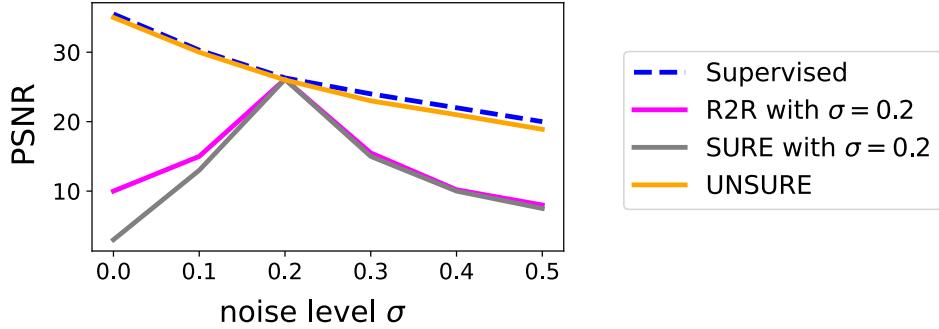


Figure 2.2: **Self-supervised denoising across noise levels** [28]. Comparison of supervised, (SURE), (R2R) and (UNSURE) losses on an MNIST Gaussian denoising task with a U-Net denoiser. If the noise level σ is correctly specified in SURE and R2R, the performance is close to the supervised case. However, if the noise level is misspecified, the performance drops significantly. The UNSURE loss is robust to noise level misspecification, and performs close to the supervised case.

both losses are very sensitive to a misspecified σ , as errors of more than 10% can result in a significant decrease of performance.

We can instead build a self-supervised loss that ensures that the divergence term in (SURE) is zero, and thus drop the dependency of the loss on the unknown noise level σ :

$$\arg \min_f \mathbb{E}_y \left\{ \frac{1}{n} \|f(\mathbf{y}) - \mathbf{y}\|^2 \right\} \text{ s.t. } \mathbb{E}_y \left\{ \sum_{i=1}^n \frac{\partial f_i}{\partial y_i}(\mathbf{y}) \right\} = 0 \quad (2.17)$$

Applying Lemma 2.4, we can show that the minimization problem is equivalent to a supervised setting with constraints:

$$\arg \min_f \mathbb{E}_{\mathbf{x}, \mathbf{y}} \left\{ \frac{1}{n} \|f(\mathbf{y}) - \mathbf{x}\|^2 \right\} \text{ s.t. } f \in \mathcal{F}$$

where we only look for denoisers that have zero-expected divergence (ZED), i.e., which belong to the constrained set

$$\mathcal{F} = \left\{ f : \mathbb{E}_y \left\{ \sum_{i=1}^n \frac{\partial f_i}{\partial y_i}(\mathbf{y}) \right\} = 0 \right\}.$$

Using a Lagrange multiplier $\eta \in \mathbb{R}$, we can rewrite the constrained learning problem in (UNSURE) as

$$\min_f \max_\eta \mathbb{E}_{\mathbf{y}} \left\{ \frac{1}{n} \|f(\mathbf{y}) - \mathbf{y}\|^2 + \frac{2\eta}{n} \sum_{i=1}^n \frac{\partial f_i}{\partial y_i}(\mathbf{y}) \right\}. \quad (\text{UNSURE})$$

Interestingly, the resulting loss is very similar to (SURE), but instead of having a fixed noise level σ , we replace it by a multiplier η and maximize over it.

Analyzing the constrained estimator What is the cost of adding the zero-expected divergence constraint on the learned denoiser? It is easy to show that the optimal denoiser has a positive divergence [74], except for the trivial case where the image distribution consists of a single image. However, we will see that the constraint is quite mild, and the gap with supervised learning can be small.

Following a similar derivation as in (Tweedie), we can obtain the optimal solution for the constrained learning problem again in terms of the score function, which can be written as

$$f^{\text{ZED}}(\mathbf{y}) = \mathbf{y} + \frac{n \nabla \log p_{\mathbf{y}}(\mathbf{y})}{\mathbb{E}_{\mathbf{y}} \|\nabla \log p_{\mathbf{y}}(\mathbf{y})\|^2}.$$

We can also compute the expected error if the zero-expected divergence estimator by simply plugging-in its definition inside (SURE):

$$\begin{aligned} \mathbb{E}_{\mathbf{y}, \mathbf{x}} \left\{ \frac{1}{n} \|f^{\text{ZED}}(\mathbf{y}) - \mathbf{x}\|^2 \right\} &= \mathbb{E}_{\mathbf{y}} \left\{ \mathcal{L}_{\text{SURE}} \left(\mathbf{y}, f^{\text{ZED}}(\mathbf{y}) \right) \right\} \\ &= \frac{n}{\mathbb{E}_{\mathbf{y}} \|\nabla \log p_{\mathbf{y}}(\mathbf{y})\|^2} - \sigma^2 \\ &= \text{MMSE} \left(1 - \frac{\text{MMSE}}{\sigma^2} \right)^{-1} \\ &= \text{MMSE} + \sigma^2 \sum_{j \geq 2} \left(\frac{\text{MMSE}}{\sigma^2} \right)^j \end{aligned} \quad (2.18)$$

where the third line uses the expression of the MMSE in (2.14) (i.e., the error of the optimal estimator $f(\mathbf{y}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{\mathbf{x}\}$) and the last line relies on the geometrical series formula. Since $\frac{\text{MMSE}}{\sigma^2}$ is the improvement in signal-to-noise ratio of the optimal estimator, we always have that $\frac{\text{MMSE}}{\sigma^2} < 1$ and generally $\frac{\text{MMSE}}{\sigma^2} \ll 1$. Thus, the additional error of f^{ZED} compared to the oracle can be quite small.

Extensions beyond isotropic Gaussian noise The (UNSURE) approach can be further extended to settings where the noise covariance is unknown, by considering an s -dimensional set of *plausible* covariance matrices

$$\mathcal{R} = \left\{ \Sigma_{\boldsymbol{\eta}} \in \mathbb{R}^{n \times n} : \Sigma_{\boldsymbol{\eta}} = \sum_{j=1}^s \eta_j \boldsymbol{\Psi}_j, \boldsymbol{\eta} \in \mathbb{R}^s \right\}$$

for some fixed basis matrices $\{\Psi_j \in \mathbb{R}^{n \times n}\}_{j=1}^s$, with the hope that the true covariance belongs to this set, that is $\Sigma \in \mathcal{R}$. In this case, we consider $s \geq 1$ constraints, and minimize the following objective

$$\arg \min_f \mathbb{E}_y \left\{ \frac{1}{n} \|f(y) - y\|^2 \right\} \quad (2.19)$$

$$\text{s.t. } \mathbb{E}_y \left\{ \text{trace} \left(\Psi_j \frac{\partial f}{\partial y}(y) \right) \right\} = 0, \quad j = 1, \dots, s \quad (2.20)$$

Note that (UNSURE) is recovered as a special case with $s = 1$ and $\Psi_1 = \mathbf{I}$. The less we know about the covariance, the larger the set \mathcal{R} , resulting in more constraints on the learned estimator. Thus, the dimension $s \geq 1$ offers a trade-off between optimality of the resulting estimator and robustness to a misspecified covariance. As with the isotropic case, we can find the closed-form expression of the optimal constrained denoiser:

Theorem 2.7 (Tachella et al. [28]). *Let $p_y = p_x * \mathcal{N}(\mathbf{0}, \Sigma)$ and assume that $\{\Psi_j \in \mathbb{R}^{n \times n}\}_{j=1}^s$ are linearly independent. The optimal solution of problem (2.19) is given by*

$$f(y) = y + \Sigma_{\hat{\eta}} \nabla \log p_y(y) \quad (2.21)$$

where the optimal multipliers are given by $\hat{\eta} = Q^{-1}v$, with

$$Q_{i,j} = \text{trace} \left(\Psi_i \mathbb{E}_y \left\{ \nabla \log p_y(y) \nabla \log p_y(y)^\top \right\} \Psi_j^\top \right)$$

and $v_i = \text{trace}(\Psi_i)$ for $i, j = 1, \dots, s$.

We can apply this generalization to problems with correlated noise and unknown correlations, which is generally modeled as

$$y = x + \sigma * \epsilon$$

where $*$ denotes the convolution operator, $\sigma \in \mathbb{R}^p$ is vector-valued and $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. If we do not know the exact noise correlation, we can consider the set of covariances with correlations up to $\pm r$ taps/pixels⁴, we can choose Σ_η to be a positive definite circulant matrix parameterized by a filter η . In this case, the solution is

$$f(y) = y + \hat{\eta} * \nabla \log p_y(y)$$

with optimal multipliers given by $2r + 1$ autocorrelation coefficients of the score [28].

We can also extend the UNSURE approach to non-Gaussian noise distributions using Lemma 2.5, such as Poisson-Gaussian noise with unknown parameters [28].

2.3.2 Cross-validation methods

In many applications, the noise is separable across pixels/measurements $p(y|x) = \prod_{i=1}^n p_i(y_i|x_i)$, but the noise distribution p_i at each pixel is unknown except for the assumption that the

⁴Here we consider 1-dimensional signals for simplicity but the result extends trivially to the 2-dimensional case.

mean is $\mathbb{E}_{y_i|x_i} \{x_i\} = x_i$. In such settings, none of the losses we presented so far are applicable, but we can still find a loss that is an unbiased estimator of a constrained supervised loss. Since we do not know the noise distribution, we need to impose *stronger constraints* than the zero expected divergence one in (UNSURE) which relies on a Gaussian noise assumption.

Many recent self-supervised methods, including Noise2Void [21], Noise2Self [75], blind spot networks [23], Neighbor2Neighbor [76], can be broadly classified as *cross-validation* approaches [47], that minimize the following objective [28]:

$$\begin{aligned} \arg \min_f \quad & \mathbb{E}_{\mathbf{y}} \left\{ \frac{1}{n} \|f(\mathbf{y}) - \mathbf{y}\|^2 \right\} \\ \text{s.t. } & \frac{\partial f_i}{\partial y_i}(\mathbf{y}) = 0, \forall \mathbf{y} \in \mathbb{R}^n, i = 1, \dots, n \end{aligned} \tag{CV}$$

where the derivative constraints are equivalent to asking that the i th output f_i doesn't depend on the i th input y_i .

Proposition 2.8 (Adapted from Batson and Royer [75]). *Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a function whose i th output does not depend on the i th entry, or equivalently, that $\frac{\partial f_i}{\partial y_i}(\mathbf{y}) = 0$ for all $\mathbf{y} \in \mathbb{R}^n$ and $i = 1, \dots, n$, and assume further that $p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^n p_i(y_i|x_i)$ and $\mathbb{E}_{y_i|x_i} \{y_i\} = x_i$ for all $i = 1, \dots, n$. Then,*

$$\mathbb{E}_{\mathbf{y}|\mathbf{x}} \{\|f(\mathbf{y}) - \mathbf{y}\|^2\} = \mathbb{E}_{\mathbf{y}|\mathbf{x}} \{\|f(\mathbf{y}) - \mathbf{x}\|^2\} + \text{const.} \tag{2.22}$$

Proof.

$$\begin{aligned} & \mathbb{E}_{\mathbf{y}|\mathbf{x}} \{\|f(\mathbf{y}) - \mathbf{y}\|^2\} \\ &= \mathbb{E}_{\mathbf{y}|\mathbf{x}} \{\|f(\mathbf{y}) - \mathbf{x}\|^2\} + 2 \sum_{i=1}^n \mathbb{E}_{\mathbf{y}|\mathbf{x}} \left\{ f_i(\mathbf{y})^\top (y_i - x_i) \right\} + \text{const.} \\ &= \mathbb{E}_{\mathbf{y}|\mathbf{x}} \{\|f(\mathbf{y}) - \mathbf{x}\|^2\} + 2 \sum_{i=1}^n \mathbb{E}_{\mathbf{y}_{-i}|\mathbf{x}} \{f_i(\mathbf{y})\}^\top \mathbb{E}_{y_i|x_i} \{y_i - x_i\} + \text{const.} \\ &= \mathbb{E}_{\mathbf{y}|\mathbf{x}} \{\|f(\mathbf{y}) - \mathbf{x}\|^2\} + \text{const.} \end{aligned}$$

where \mathbf{y}_{-i} denotes the a vector with all entries of \mathbf{y} except for the i th entry, the third line uses the fact that $f_i(\mathbf{y})$ and y_i are independent conditional on \mathbf{x} , and the last line uses $\mathbb{E}_{y_i|x_i} \{y_i\} = x_i$. \square

Due to Proposition 2.8, the minimization problem in (CV) is equivalent to the following constrained supervised problem

$$\arg \min_f \mathbb{E}_{\mathbf{x}, \mathbf{y}} \left\{ \frac{1}{n} \|f(\mathbf{y}) - \mathbf{x}\|^2 \right\} \quad \text{s.t. } f \in \mathcal{F}$$

where we only look for denoisers which do not use the i th input value for predicting the i th output value, that is

$$\mathcal{F} = \left\{ f : \frac{\partial f_i}{\partial y_i}(\mathbf{y}) = 0, \forall \mathbf{y} \in \mathbb{R}^n, i = 1, \dots, n \right\}.$$

As with (UNSURE), the conditional mean $f^*(\mathbf{y}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}}\{\mathbf{x}\}$ does not belong to the constrained set \mathcal{F} , and we cannot expect to achieve the same performance as supervised learning. In this case, the optimal solution of (CV) is $f_i^{\text{CV}}(\mathbf{y}) = \mathbb{E}_{x_i|\mathbf{y}_{-i}}\{x_i\}$ for $i = 1, \dots, n$. Here, unlike the UNSURE case where the suboptimality gap is available closed form in (2.18), the gap between the f^* and f^{CV} does not admit a simple closed-form expression and will be highly dependent on the signal distribution: we expect the gap to be smaller in signals exhibiting strong spatial correlations, and larger in sparse signals [75].

Two main approaches have been proposed for enforcing the zero derivatives constraints: (1) blind spot networks, which use a specific architecture that enforce the constraint, and (2) splitting losses, which enforce it during training.

Blind-spot networks Laine et al. [23] proposed an image-to-image network architecture f which only relies on the neighbors of a pixel (and not the pixel itself) to predict its denoised value, which is equivalent to imposing $\frac{\partial f_i}{\partial y_i}(\mathbf{y}) = 0$ for all pixels $i = 1, \dots, n$. This blind-spot network relies on a fully convolutional architecture which combines shifted (upwards, downwards, leftwards and rightwards) receptive fields. This approach is very efficient since it allows one to train the denoiser f directly on the measurement consistency loss $\|f(\mathbf{y}) - \mathbf{y}\|^2$, but imposes strong architectural constraints on f above and beyond the zero gradient constraint.

Splitting methods A second approach to training networks that do not rely on the central pixel for denoising is to randomly mask this pixel out during the training procedure [21, 75]. This approach can be written as the following loss

$$\mathcal{L}_{\text{SPLIT}}(\mathbf{y}, f) = \mathbb{E}_{\mathbf{b}} \left\{ \frac{1}{n} \|(\mathbf{1} - \mathbf{b}) \circ (f(\mathbf{b} \circ \mathbf{y}) - \mathbf{y})\|^2 \right\} \quad (\text{SPLIT})$$

where \circ denotes the elementwise (Hadamard) product and $\mathbf{b} \in \{0, 1\}^n$ are random binary masks. There is a large literature regarding the choice of splitting distribution $p(\mathbf{b})$, which generally depends on the structure of the data (e.g., images, videos, etc.). Some of the most popular choices are:

1. Noise2Void [21] replaces a central input pixel with a random neighboring pixel, and computes the loss only on the pixels that have been replaced.
2. Neighbor2Neighbor [76] constructs two non-overlapping subsampled versions of an image by randomly choosing a pixel from every 2×2 patch for the input and another pixel from the same patch for the target. This loss implicitly assumes that the images are scale-invariant, and thus the denoiser can be trained on undersampled images and tested at full resolution.
3. Noise2Self [75] partitions an image using J disjoint masks, $\mathbf{b}^{(j)}$, such that $\frac{1}{J} \sum_{j=1}^J b_i^{(j)} = 1$ for $i = 1, \dots, n$, and then trains an estimator, f , constructed in the following manner:

$$f(\mathbf{y}) = \frac{1}{J} \sum_{j=1}^J \mathbf{b}^{(j)} \circ g((1 - \mathbf{b}^{(j)}) \circ \mathbf{y} + \mathbf{b}^{(j)} \circ \mathbf{u})$$

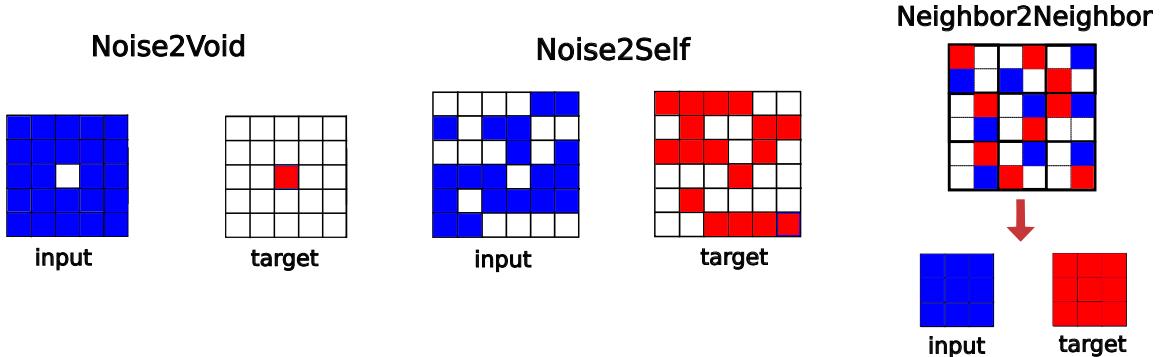


Figure 2.3: **Pixel splitting strategies.** Noise2Void and Noise2Self zero-fill or copy neighboring values to the pixels removed from the input image, whereas Neighbor2Neighbor splits using two random subsamplings of every 2×2 neighborhood, one as input and the other as the target.

where g is a based neural network and \mathbf{u} is an i.i.d. uniformly distributed random vector⁵.

Figure 2.3 illustrates the different masking approaches. It is also possible to use a more general collection of J random masks, $\mathbf{b}^{(j)}$, as long as they cover the entire image (that is $\sum_{s=1}^J (1 - b_i^{(s)}) > 0$ for all pixels $i = 1, \dots, n$). Then, in a similar fashion to (R2R), at test time we can average over multiple splittings [77]:

$$f^{\text{test}}(\mathbf{y}) = \sum_{j=1}^J \mathbf{w}^{(j)} \circ f(\mathbf{b}^{(j)} \circ \mathbf{y}) \quad (2.23)$$

with weights

$$w_i^{(j)} = \frac{1 - b_i^{(j)}}{\sum_{s=1}^J (1 - b_i^{(s)})} \quad \text{for } i = 1, \dots, n$$

In the next chapter, we will present an extension of this idea to general inverse problems, where the forward operator is non-trivial $\mathbf{A}(\mathbf{x}) \neq \mathbf{x}$, and splitting strategies are developed in an operator-specific way.

2.4 Learning generative models from noisy data

We have seen that approximating the posterior mean is possible with a self-supervised loss, as long as the noise distribution is known. We can also ask whether other posterior statistics beyond the mean can be approximated, or more generally, if we can estimate the signal distribution, $p_{\mathbf{x}}$, from measurement data alone. Since we have that

$$p_{\mathbf{y}}(\mathbf{y}) = \int_{\mathbf{x}} p(\mathbf{y}|\mathbf{x})p_{\mathbf{x}}(\mathbf{x})d\mathbf{x} \quad (2.24)$$

⁵Although not theoretically justified, [75] also observed good performance by simply using the trained based neural network, g , as the final estimator.

this problem can be seen as a linear inverse problem in the space of measures, which can be written as $p_{\mathbf{y}} = \mathcal{A}(p_{\mathbf{x}})$, where \mathcal{A} is associated to the integral with a kernel $p(\mathbf{y}|\mathbf{x})$. This formulation can be traced back to Robbins work [78] in empirical Bayes estimators.

Here we consider the simplest setup: a denoising problem with additive noise, i.e., $\mathbf{y} = \mathbf{x} + \boldsymbol{\epsilon}$. Since the noise is additive, we have that the measurement distribution is a convolved version of the signal distribution, i.e.,

$$p_{\mathbf{y}} = p_{\mathbf{x}} * p_{\boldsymbol{\epsilon}} \quad (2.25)$$

and thus model identification can be seen as a deconvolution problem. The Fourier analog of this problem is

$$\phi_{\mathbf{y}}(\boldsymbol{\omega}) = \phi_{\mathbf{x}}(\boldsymbol{\omega})\phi_{\boldsymbol{\epsilon}}(\boldsymbol{\omega}) \quad (2.26)$$

where $\phi_{\mathbf{y}}(\boldsymbol{\omega}) = \mathbb{E}_{\mathbf{y}}\{\exp(i\mathbf{y}^T \boldsymbol{\omega})\}$, $\phi_{\mathbf{x}}(\boldsymbol{\omega}) = \mathbb{E}_{\mathbf{x}}\{\exp(i\mathbf{x}^T \boldsymbol{\omega})\}$ and $\phi_{\boldsymbol{\epsilon}}(\boldsymbol{\omega}) = \mathbb{E}_{\boldsymbol{\epsilon}}\{\exp(i\boldsymbol{\epsilon}^T \boldsymbol{\omega})\}$ are the characteristic functions of $p_{\mathbf{x}}$, $p_{\mathbf{y}}$ and $p_{\boldsymbol{\epsilon}}$ and $\boldsymbol{\omega} \in \mathbb{R}^n$. By simple inspection of (2.26), we can deduce that, if the noise model is known and hence $\phi_{\boldsymbol{\epsilon}}(\boldsymbol{\omega})$ is known, it is possible to identify the signal distribution as long as $\phi_{\boldsymbol{\epsilon}}(\boldsymbol{\omega}) \neq 0$ for all $\boldsymbol{\omega} \in \mathbb{R}^n$:

Proposition 2.9 (Tachella et al. [79]). *If the characteristic function of the noise distribution $\phi_{\boldsymbol{\epsilon}}$ is nowhere zero, then there is a one-to-one mapping between the spaces of clean measurement distributions and noisy measurement distributions.*

For example, the Gaussian noise distribution has a nowhere zero characteristic function, and we can thus uniquely identify the signal distribution from noisy measurements⁶. We now present some recent methods aiming to learn a generative model from noisy data alone.

Variational autoencoders Assuming that the noise model $p(\mathbf{y}|\mathbf{x})$ is known (or estimated in a calibration step), we can model the distribution of measurement data as

$$\begin{cases} \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \hat{\mathbf{x}} = f(\mathbf{z}) \\ \hat{\mathbf{y}} \sim p(\mathbf{y}|\mathbf{x} = \hat{\mathbf{x}}) \end{cases} \quad (2.27)$$

where f is a deep network, and \mathbf{z} are latent variables that follow a standard Gaussian distribution. Prakash et al. [80, 81] propose to learn such a generative model using a variational autoencoder, which requires learning an additional encoder network to approximate the distribution $p(\mathbf{z}|\mathbf{y})$. Once both encoder, $p(\mathbf{z}|\mathbf{y})$, and decoder, f , are learned, at inference we can either generate clean samples from $p_{\mathbf{x}}$ as $\hat{\mathbf{x}}^{(i)} = f(\mathbf{z}^{(i)})$ for $i = 1, \dots, N$ where $\mathbf{z}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, or generate clean samples from the posterior distribution by first sampling the latent variables from the encoder, $\mathbf{z}^{(i)} \sim p(\mathbf{z}|\mathbf{y})$ and then passing these through the decoder network. In the next chapter, we will present adversarial methods [82] for learning a similar generative model in the case of incomplete measurements.

Diffusion methods (Tweedie) shows that optimal Gaussian denoisers as a function of noise level provide access to the score function of the noisy signal distribution, $\nabla \log p_{\mathbf{y}}(\mathbf{y})$. Diffusion models leverage this using neural network approximations to the score to build

⁶Note that this says nothing about the sample complexity of this problem, i.e., how hard this would be from finite data.

stochastic samplers that can approximately sample from the clean signal distribution $p(\mathbf{x})$ or approximately sample from the conditional posterior $p(\mathbf{x}|\mathbf{y})$ density given a noisy instance, \mathbf{y} [41].

As we have seen in this chapter, we can learn the denoisers in a self-supervised manner via (SURE) or (R2R) using noisy data alone. However, this only gives us access to an approximation for the score at a noise level greater than or equal to the observed data, $\sigma \geq \sigma_n$. Some recent approaches stop the diffusion at this noise level [83], while others attempt to go below this by imposing a consistency constraint on the learned denoiser [44].

2.5 Towards general inverse problems

What happens if we want to extend the losses in this chapter beyond a simple denoising problem, where $\mathbf{A} : \mathbb{R}^n \mapsto \mathbb{R}^m$ is a general forward operator? Defining a denoising function as $\mathbf{A} \circ f$, we can use any of the losses above in the measurement space. For example, if we observe measurements with Gaussian noise $\mathbf{y} \sim \mathcal{N}(\mathbf{A}(\mathbf{x}), \sigma^2 \mathbf{I})$ where the noise level σ is known, we can adapt (SURE) as

$$\mathcal{L}_{\text{SURE}}(\mathbf{y}, f) = \frac{1}{n} \|\mathbf{y} - \mathbf{A} \circ f(\mathbf{y})\|^2 + \frac{2}{n} \text{trace} \left(\Sigma \frac{\partial \mathbf{A} \circ f}{\partial \mathbf{y}}(\mathbf{y}) \right)$$

so that we have the equivalent to the following measurement supervised loss

$$\mathbb{E}_{\mathbf{y}} \{ \mathcal{L}_{\text{SURE}}(\mathbf{y}, f) \} = \mathbb{E}_{\mathbf{x}, \mathbf{y}} \left\{ \frac{1}{n} \|\mathbf{A}(\mathbf{x}) - \mathbf{A} \circ f(\mathbf{y})\|^2 \right\} + \text{const.}$$

and importantly the minimizer is $f^*(\mathbf{y}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ \mathbf{x} \}$ if \mathbf{A} is a one-to-one mapping⁷. However, if \mathbf{A} is not one-to-one, for example if there are more pixels n than measurements m , then $f^*(\mathbf{y}) \neq \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ \mathbf{x} \}$, and we cannot expect to learn the same solution as in the supervised setting, even if we have a dataset with infinitely many measurement vectors. The next chapter will present some solutions to overcome this limitation.

2.6 Summary

In this chapter, we have seen how to build self-supervised losses that can handle noisy data without requiring access to clean targets. The choice of the loss is dependent on how much knowledge we have about the noise distribution: the more we know, the closer we can expect to get to the performance of fully supervised learning, the less we know, the more constraints we need to impose on the learned denoiser and the further we get from the supervised performance. Nonetheless, we have seen that in many cases, the gap with supervised learning can be small, and self-supervised losses can be used to train denoisers that perform well in practice. Section 2.6 shows a summary of the different loss families covered in this chapter, highlighting the different noise assumptions of each loss.

⁷In this case, the self-supervised loss will share the same minimizer as the supervised loss, i.e. satisfying (1.10).

Loss family	Noise assumption	Learns optimal $f^*(\mathbf{y}) = \mathbb{E}_{\mathbf{x} \mathbf{y}} \{\mathbf{x}\}$?	Refs.
(Noise2Noise)	Two independent noise realizations $(\mathbf{y}_1, \mathbf{y}_2)$	Yes	[40]
(R2R) (GR2R)	Natural exponential family or additive noise known parameters	Yes	[51, 52] [54, 55]
(SURE)	Natural exponential family and Poisson-Gaussian known parameters	Yes	[53, 63] [60, 67] [61, 62]
(UNSURE)	Natural exponential family and Poisson-Gaussian unknown parameters	No, but small gap see (2.18)	[28]
(CV)	$p(\mathbf{y} \mathbf{x}) = \prod_{i=1}^n p_i(y_i x_i)$	No, gap depends on spatial corr.	[75, 76] [21, 23] [84]

Table 2.2: **Summary of the self-supervised losses covered in this chapter.** The natural exponential family includes many popular noise distributions, such as Gaussian, Poisson and Gamma. All losses assume that the noise verifies $\mathbb{E}_{\mathbf{y}|\mathbf{x}} \{\mathbf{y}\} = \mathbf{x}$ (or $\mathbb{E}_{\mathbf{y}_1|\mathbf{x}} \{\mathbf{y}_1\} = \mathbf{x}$ for Noise2Noise).

Chapter 3

Learning from incomplete measurements

The self-supervised losses presented in the previous chapter can handle various types of noise model and are applicable to any one-to-one forward operator. However, what happens when the operator is many-to-one, i.e., non-invertible? It is easy to show that, even in the simple case of noiseless measurements and linear operator $\mathbf{A}(\mathbf{x}) = \mathbf{Ax}$, we do not have any information in the nullspace of \mathbf{A} , the linear subspace $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{0}\}$, and thus we cannot learn the reconstruction function in this part of the space:

Proposition 3.1 (Chen et al. [26]). *Any reconstruction function $f : \mathbb{R}^m \mapsto \mathbb{R}^n$ of the form*

$$f(\mathbf{y}) = \mathbf{A}^\dagger \mathbf{y} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A}) v(\mathbf{y}) \quad (3.1)$$

where \mathbf{A}^\dagger is the pseudo-inverse of \mathbf{A} and $v : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is any function, verifies measurement consistency $\mathbf{Af}(\mathbf{y}) = \mathbf{y}$, and thus is a global minimizer of $\mathbb{E}_{\mathbf{y}} \left\{ \frac{1}{m} \|\mathbf{Af}(\mathbf{y}) - \mathbf{Ax}\|^2 \right\}$.

In order to tackle this problem we therefore need additional information. There are two main ways to overcome this limitation: the first one, covered in Section 3.1, is to train with measurements associated to different forward operators $\{\mathbf{A}_g\}_{g=1}^G$, each with possibly a different nullspace. The second option, covered in Section 3.2, is to assume some invariance of the set of images to a group of transformations $\{\mathbf{T}_g\}_{g=1}^G$, which we will show that is equivalent to observing measurements with the set of operators $\{\mathbf{A} \circ \mathbf{T}_g\}_{g=1}^G$.

3.1 Leveraging multiple operators

We assume that measurements are obtained via the following model

$$\begin{cases} \mathbf{x} \sim p(\mathbf{x}), \mathbf{A} \sim p(\mathbf{A}) \\ \mathbf{y} \sim p(\mathbf{y} | \mathbf{Ax}) \end{cases} \quad (3.2)$$

where we consider only *linear forward operators* and assume that a different operator \mathbf{A} is sampled for every measurement \mathbf{y} . In practice, the distribution of operators $p(\mathbf{A})$ is generally discrete and finite. Some practical examples are (see Figure 3.1):

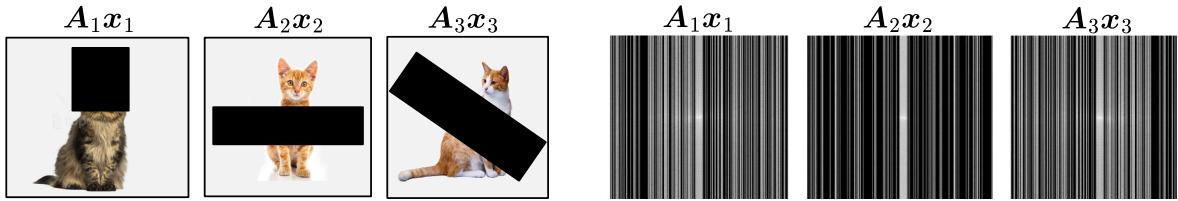


Figure 3.1: **Learning with multiple forward operators.** In some imaging settings, such as image inpainting (left) and accelerated MRI (right), the masking operator changes across samples in the dataset, offering different views of the signal distribution.

- In accelerated MRI applications [27], the acceleration mask might change from scan to scan, where $\mathbf{A} = \text{diag}(\mathbf{b}) \mathbf{F}$ where \mathbf{F} is the discrete Fourier transform and the mask is randomly sampled as $\mathbf{b} \sim p(\mathbf{b})$.
- In image inpainting problems, $\mathbf{A} = \text{diag}(\mathbf{b})$, the missing pixels often vary from image to image [51, 82], resulting in a set of different masks.

Information from multiple operators can help us overcome the limitation of Proposition 3.1. Imagine that there are a maximum set of G forward operators to draw from, $p(\mathbf{A}) = \frac{1}{G} \sum_{g=1}^G \delta_{\mathbf{A}_g}$, and the signal distribution is composed of a single signal, $p(\mathbf{x}) = \delta_{\mathbf{x}_0}$. In this trivial case, we would have access to G different measurement vectors, $\mathbf{y}_g = \mathbf{A}_g \mathbf{x}_0$ for $g = 1, \dots, G$, all measuring the same underlying signal \mathbf{x}_0 , each via one out of the G different forward operators $\mathbf{A}_g \in \mathbb{R}^{m \times n}$. We could then try to recover \mathbf{x}_0 by solving the following system of equations:

$$\begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_G \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 \\ \vdots \\ \mathbf{A}_G \end{bmatrix} \mathbf{x}_0 \quad (3.3)$$

which has mG equations and n unknowns. As this is the maximum number of measurements that we can obtain, to be able to identify \mathbf{x}_0 from the measurement data, it is necessary that the system in (3.3) has maximal rank, or equivalently that $\mathbb{E}_{\mathbf{A}} \{ \mathbf{A}^\top \mathbf{A} \} = \frac{1}{G} \sum_{g=1}^G \mathbf{A}_g^\top \mathbf{A}_g$ is an invertible matrix.

This argument gives us a necessary condition on the number and diversity of operators required to learn from incomplete measurements in the general case of non trivial signal distribution. It tells us that we need at least $G \geq n/m$ different forward operators in order to learn from incomplete measurements. However, it does not constitute a practical algorithm nor provides a sufficient condition in the general case of a non-trivial signal distribution, since there will typically be zero probability of observing the same image more than once.

Assuming the operators are known, we can make explicit the dependency of the reconstruction network on the forward operator as $f(\mathbf{y}, \mathbf{A})$. As discussed in the first chapter, there are various ways to incorporate knowledge of the forward operator into the architecture, with the most common being unrolled optimization algorithms, e.g., [14, 15]. A naive approach for handling multiple operators is to minimize measurement consistency:

$$\arg \min_f \mathbb{E}_{\mathbf{y}, \mathbf{A}} \{ \mathcal{L}_{\text{MC}}(\mathbf{y}, \mathbf{A} \circ f) \} \quad (3.4)$$

with

$$\mathcal{L}_{\text{MC}}(\mathbf{y}, \mathbf{A} \circ f) = \frac{1}{m} \|\mathbf{A}f(\mathbf{y}, \mathbf{A}) - \mathbf{y}\|^2. \quad (\text{MC})$$

Unfortunately, this idea might fail, as $f(\mathbf{y}, \mathbf{A}) = \mathbf{A}^\dagger \mathbf{y}$ is a global minimizer of the loss. This trivial solution is due to the fact that f can achieve zero training error without learning anything about the signal distribution. We can avoid this solution in two ways, using a splitting loss or enforcing consistency across the operators.

Splitting with noiseless measurements We can avoid the trivial solution of the measurement consistency loss by removing some of the measurements from the input, such that f needs to predict the unobserved part. Dividing our measurements into two non-overlapping parts, i.e., $\mathbf{y} = [\mathbf{y}_1^\top, \mathbf{y}_2^\top]^\top$ and $\mathbf{A}^\top = [\mathbf{A}_1^\top, \mathbf{A}_2^\top]^\top$, we can build a self-supervised loss asking the network to predict \mathbf{y} given \mathbf{y}_1 :

$$\mathcal{L}_{\text{MSPLIT}}(\mathbf{y}, \mathbf{A}, f) = \mathbb{E}_{\mathbf{y}_1, \mathbf{A}_1 | \mathbf{y}, \mathbf{A}} \left\{ \frac{1}{m} \|\mathbf{A}f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{y}\|^2 \right\} \quad (\text{MSPLIT})$$

where the expectation averages over some distribution $p(\mathbf{y}_1, \mathbf{A}_1 | \mathbf{y}, \mathbf{A})$ of random splittings. In practice, a single splitting is sampled per gradient step when training a network. There is an extensive literature on how to choose the splitting distribution, which is generally problem-dependent. For example, in accelerated MRI, a popular strategy [27] is to split acceleration masks into non-overlapping sub masks, generally leaving most of the low-frequency information in \mathbf{A}_1 to avoid losing too much information.

It is easy to verify that the trivial solution $f(\mathbf{y}_1, \mathbf{A}_1) = \mathbf{A}_1^\dagger \mathbf{y}_1$ is not a global minimizer of the expected (MSPLIT) loss. An important question is then, does the loss approximate the supervised loss? Assuming the observation model in (3.2) with noiseless measurements $\mathbf{y} = \mathbf{Ax}$, we have the following result:

Proposition 3.2 (Adapted from Daras et al. [85]). *Assume the observation model given by (3.2) with noiseless measurements, i.e., $\mathbf{y} = \mathbf{Ax}$. The multioperator splitting loss is an unbiased estimator of a weighted supervised loss, that is*

$$\begin{aligned} & \mathbb{E}_{\mathbf{y}, \mathbf{A}} \{ \mathcal{L}_{\text{MSPLIT}}(\mathbf{y}, \mathbf{A}, f) \} \\ &= \mathbb{E}_{\mathbf{y}_1, \mathbf{A}_1, \mathbf{x}} \left\{ (f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{x})^\top \mathbf{Q}_{\mathbf{A}_1} (f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{x}) \right\} \end{aligned}$$

where

$$\mathbf{Q}_{\mathbf{A}_1} = \mathbb{E}_{\mathbf{A} | \mathbf{A}_1} \left\{ \mathbf{A}^\top \mathbf{A} \right\} \quad (3.5)$$

and the global minimizers of the expected loss are given by

$$f^*(\mathbf{y}_1, \mathbf{A}_1) = \mathbf{Q}_{\mathbf{A}_1}^\dagger \mathbf{Q}_{\mathbf{A}_1} \mathbb{E}_{\mathbf{x} | \mathbf{y}_1, \mathbf{A}_1} \{ \mathbf{x} \} + (\mathbf{I} - \mathbf{Q}_{\mathbf{A}_1}^\dagger \mathbf{Q}_{\mathbf{A}_1}) v(\mathbf{y}_1) \quad (3.6)$$

where $v : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is any function.

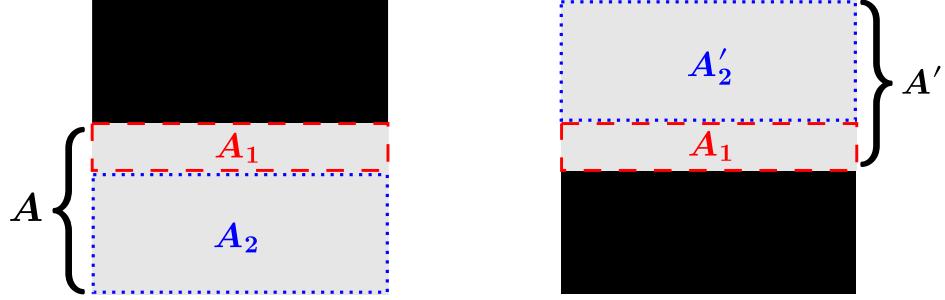


Figure 3.2: **Illustration of Proposition 3.2** Consider a simple image inpainting problem $\mathbf{A} = \text{diag}(\mathbf{b})$ for some mask \mathbf{b} , where images are measured by one of the two masks in gray at random, $\mathbf{b} \sim p(\mathbf{b})$. The \mathbf{A}_1 split denoted by red dashed lines is present in both masks, and thus we have that $\mathbf{Q}_{\mathbf{A}_1} = \mathbf{A}^\dagger \mathbf{A} + (\mathbf{A}')^\dagger \mathbf{A}'$ is the identity matrix. Hence, the global minimizer of the expected splitting loss is the conditional mean $\mathbb{E}_{\mathbf{x}|\mathbf{A}_1 \mathbf{x}} \{\mathbf{x}\}$.

Proof.

$$\mathbb{E}_{\mathbf{y}, \mathbf{A}} \{\mathcal{L}_{\text{MSPLIT}}(\mathbf{y}, \mathbf{A}, f)\} \quad (3.7)$$

$$= \mathbb{E}_{\mathbf{y}_1, \mathbf{A}_1, \mathbf{y}, \mathbf{A}} \{\|\mathbf{A}f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{y}\|^2\} \quad (3.8)$$

$$= \mathbb{E}_{\mathbf{y}_1, \mathbf{A}_1, \mathbf{x}} \left\{ \mathbb{E}_{\mathbf{A}|\mathbf{A}_1} \left\{ \frac{1}{m} \|\mathbf{A}f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{A}\mathbf{x}\|^2 \right\} \right\} \quad (3.9)$$

$$= \mathbb{E}_{\mathbf{y}_1, \mathbf{A}_1, \mathbf{x}} \left\{ \frac{1}{m} (f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{x})^\top \mathbf{Q}_{\mathbf{A}_1} (f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{x}) \right\} \quad (3.10)$$

where the second line uses the definition of the observation model, the third line relies on the noiseless measurements assumption and the last line groups uses the definition of $\mathbf{Q}_{\mathbf{A}_1}$. Since this is a weighted ℓ_2 loss, we can apply Proposition C.2 to conclude that any minimizer of this loss is given by (3.6). \square

If the matrix $\mathbf{Q}_{\mathbf{A}_1}$ is invertible for some split \mathbf{A}_1 , then $\mathbf{Q}_{\mathbf{A}_1}^\dagger \mathbf{Q}_{\mathbf{A}_1} = \mathbf{I}$, and the minimizer in expectation (c.f. (1.10)) is unique, and is given by the conditional mean

$$f^*(\mathbf{y}_1, \mathbf{A}_1) = \mathbb{E}_{\mathbf{x}|\mathbf{y}_1, \mathbf{A}_1} \{\mathbf{x}\}. \quad (3.11)$$

Having an invertible $\mathbf{Q}_{\mathbf{A}_1}$ is thus a sufficient condition to obtain a conditional mean estimator similar to the supervised case, but it is not necessary, since it relies on a single split \mathbf{A}_1 of the observed matrix \mathbf{A} , whereas we could average over all J possible splittings $\{(\mathbf{y}_1^{(j)}, \mathbf{A}_1^{(j)})\}_{j=1}^J$ of the observed (\mathbf{y}, \mathbf{A}) at test time. If the average

$$\bar{\mathbf{Q}}_{\mathbf{A}} = \frac{1}{J} \sum_{j=1}^J \mathbf{Q}_{\mathbf{A}_1^{(j)}} = \mathbb{E}_{\mathbf{A}_1|\mathbf{A}} \{\mathbf{Q}_{\mathbf{A}_1}\} \quad (3.12)$$

is invertible, we can compute the average prediction at test time as

$$f^{\text{test}}(\mathbf{y}, \mathbf{A}) = \frac{1}{J} \sum_{j=1}^J \bar{\mathbf{Q}}_{\mathbf{A}}^{-1} \mathbf{Q}_{\mathbf{A}_1^{(j)}} f(\mathbf{y}_1^{(j)}, \mathbf{A}_1^{(j)}) \quad (3.13)$$

where $\bar{\mathbf{Q}}_{\mathbf{A}}^{-1} \mathbf{Q}_{\mathbf{A}_1^{(j)}}$ can be seen as weighting terms that sum up to the identity, $\frac{1}{J} \sum_{j=1}^J \bar{\mathbf{Q}}_{\mathbf{A}}^{-1} \mathbf{Q}_{\mathbf{A}_1^{(j)}} = \mathbf{I}$. Applying Proposition 3.2, we can show that the test time estimate approximates the following average of conditional means:

$$\begin{aligned} & f^{\text{test}}(\mathbf{y}, \mathbf{A}) \\ & \approx \frac{1}{J} \sum_{j=1}^J \bar{\mathbf{Q}}_{\mathbf{A}}^{-1} \mathbf{Q}_{\mathbf{A}_1^{(j)}} \left(\mathbf{Q}_{\mathbf{A}_1^{(j)}}^\dagger \mathbf{Q}_{\mathbf{A}_1^{(j)}} \mathbb{E}_{\mathbf{x}|\mathbf{y}_1, \mathbf{A}_1^{(j)}} \{ \mathbf{x} \} + (\mathbf{I} - \mathbf{Q}_{\mathbf{A}_1^{(j)}}^\dagger \mathbf{Q}_{\mathbf{A}_1^{(j)}}) v(\mathbf{y}_1) \right) \\ & = \frac{1}{J} \sum_{j=1}^J \bar{\mathbf{Q}}_{\mathbf{A}}^{-1} \mathbf{Q}_{\mathbf{A}_1^{(j)}} \mathbb{E}_{\mathbf{x}|\mathbf{y}_1, \mathbf{A}_1^{(j)}} \{ \mathbf{x} \} \\ & = \mathbb{E}_{\mathbf{A}_1|\mathbf{A}} \left\{ \bar{\mathbf{Q}}_{\mathbf{A}}^{-1} \mathbf{Q}_{\mathbf{A}_1} \mathbb{E}_{\mathbf{x}|\mathbf{y}_1, \mathbf{A}_1} \{ \mathbf{x} \} \right\}. \end{aligned}$$

In general, closed-form expressions of $\mathbf{Q}_{\mathbf{A}_1}$ and $\bar{\mathbf{Q}}_{\mathbf{A}}$ are not tractable, and practitioners often use a single split or multiple equally-weighted splits $f^{\text{test}}(\mathbf{y}, \mathbf{A}) = \frac{1}{J} \sum_{j=1}^J f(\mathbf{y}_1^{(j)}, \mathbf{A}_1^{(j)})$ at test time, albeit without any theoretical guarantees. However, in some specific cases, such as diagonal operators [86], it is possible to compute them explicitly, as illustrated in the example below.

Example 3.3. Consider an image inpainting problem $\mathbf{A} = \text{diag}(\mathbf{b})$ with random masks, where $b_i \sim \text{Ber}(p_i)$ taking values in $\{0, 1\}$ for $i = 1, \dots, n$. We can split measurements by an additional masking operation, such that $\mathbf{A}_1 = \text{diag}(\mathbf{b}_1)$ and $\mathbf{A}_2 = \text{diag}(\mathbf{b}_2)$ are also masking operators with $\mathbf{b}_1 = \mathbf{b} \circ \boldsymbol{\omega}$ and $\mathbf{b}_2 = \mathbf{b} \circ (\mathbf{1} - \boldsymbol{\omega})$ with splitting mask sampled as $\omega_i \sim \text{Ber}(q_i)$. In this case, we have that both $\mathbf{Q}_{\mathbf{A}_1}$ and $\bar{\mathbf{Q}}_{\mathbf{A}}$ are diagonal matrices, as they are given by averages over diagonal matrices. Due to the separability across pixels of the mask sampling distributions, we can focus the analysis on a single entry $i \in \{1, \dots, n\}$. Letting $[\mathbf{A}_2]_{i,i} = b_i(1 - \omega_i)$ with fixed $[\mathbf{A}_1]_{i,i} = b_{1,i}$, the diagonal entries of $\mathbf{Q}_{\mathbf{A}_1}$ are given by

$$[\mathbf{Q}_{\mathbf{A}_1}]_{i,i} = \mathbb{E}_{b_i, \omega_i | b_i \omega_i = b_{1,i}} \{ b_i \}$$

Since $\mathbb{E}_{b_i, \omega_i | b_i \omega_i = 0} \{ b_i \} = \frac{p_i(1-q_i)}{1-p_i q_i}$ and $\mathbb{E}_{b_i, \omega_i | b_i \omega_i = 1} \{ b_i \} = 1$ we have that

$$[\mathbf{Q}_{\mathbf{A}_1}]_{i,i} = \begin{cases} 1 & \text{if } b_{1,i} = 1 \\ \frac{p_i(1-q_i)}{1-p_i q_i} & \text{otherwise} \end{cases}.$$

Thus, if $p_i > 0$ and $q_i < 1$ for all $i = 1, \dots, n$, then $\mathbf{Q}_{\mathbf{A}_1}$ is invertible for all splits \mathbf{A}_1 .

We can now compute $\bar{\mathbf{Q}}_{\mathbf{A}}$. For a given \mathbf{A} with $[\mathbf{A}]_{i,i} = b_i$, the diagonal entries of $\bar{\mathbf{Q}}_{\mathbf{A}}$ are given by

$$[\bar{\mathbf{Q}}_{\mathbf{A}}]_{i,i} = \begin{cases} \frac{p_i(1-q_i)}{1-p_i q_i} & \text{if } b_i = 0 \\ \frac{p_i(1-q_i)^2}{1-p_i q_i} & \text{if } b_i = 1 \end{cases}.$$

Again, if $p_i > 0$ and $q_i < 1$ for all $i = 1, \dots, n$ to have an invertible $\bar{\mathbf{Q}}_{\mathbf{A}}$ for all possible inpainting masks¹ \mathbf{A} [51].

¹Although presented here as a multiple operator inpainting problem, it was originally proposed in [51] as a multiplicative version of the Noiser2Noise self-supervised denoising technique.

This example also holds for problems with $\mathbf{A} = \text{diag}(\mathbf{b}) \mathbf{F}$ where \mathbf{F} is a fixed invertible matrix, such as the discrete Fourier transform in accelerated MRI [77], and diagonal values are sampled as $\mathbf{b} \sim p(\mathbf{b})$. Moreover, in these problems, if $\mathbf{Q}_{\mathbf{A}_1}$ is invertible for all possible splits \mathbf{A}_1 , it is possible to consider a weighted version of the splitting loss [77]:

$$\mathcal{L}_{\text{MSPLIT}}(\mathbf{y}, \mathbf{A}, f) = \mathbb{E}_{\mathbf{y}_1, \mathbf{A}_1 | \mathbf{y}, \mathbf{A}} \left\{ \frac{1}{m} \|\mathbf{Q}_{\mathbf{A}_1}^{-\frac{1}{2}} (\mathbf{A}f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{y})\|^2 \right\}$$

to obtain an unbiased estimate of the supervised loss, that is

$$\mathbb{E}_{\mathbf{y}, \mathbf{A}} \{\mathcal{L}_{\text{MSPLIT}}(\mathbf{y}, \mathbf{A}, f)\} = \mathbb{E}_{\mathbf{y}_1, \mathbf{A}_1} \left\{ \frac{1}{n} \|f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{x}\|^2 \right\} + \text{const.}$$

This additional weighting does not modify the global minimizer of the expected loss, but it can improve the performance of the learned f in practice where the dataset is finite.

Splitting with noisy measurements What happens if the measurements have noise? We can analyze this case by decomposing the (MSPLIT) loss as

$$\mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2, \mathbf{A}_1, \mathbf{A}_2 | \mathbf{y}, \mathbf{A}} \left\{ \mathcal{L}_{\text{MC}}(\mathbf{y}_1, \mathbf{A}_1 \circ f) + \frac{1}{m} \|\mathbf{A}_2 f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{y}_2\|^2 \right\}$$

where the first term penalizes the measurement consistency with the input \mathbf{y}_1 as in (MC), and the second term is associated with the prediction of the unobserved part, \mathbf{y}_2 . If we have noisy measurement data that is separable across measurements, i.e., $p(\mathbf{y} | \mathbf{A}\mathbf{x}) = \prod_{i=1}^n p(y_i | \mathbf{a}_i^\top \mathbf{x})$ where $\mathbf{a}_i \in \mathbb{R}^n$ denotes the i th row of \mathbf{A} , the second term is equivalent to the noiseless case, as \mathbf{y}_1 and \mathbf{y}_2 are independent given the underlying signal \mathbf{x} , that is

$$\mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2} \{\|\mathbf{A}_2 f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{y}_2\|^2\} = \mathbb{E}_{\mathbf{y}_1, \mathbf{x}} \{\|\mathbf{A}_2 f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{A}_2 \mathbf{x}\|^2\}$$

due to Proposition 2.1.

However, the measurement consistency term is not equivalent to the noiseless case, as the noise is the same in both input and target:

$$\mathbb{E}_{\mathbf{y}_1, \mathbf{A}_1} \{\mathcal{L}_{\text{MC}}(\mathbf{y}_1, \mathbf{A}_1 \circ f)\} \neq \mathbb{E}_{\mathbf{y}_1, \mathbf{A}_1, \mathbf{x}} \left\{ \frac{1}{m} \|\mathbf{A}_1 f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{A}_1 \mathbf{x}\|^2 \right\}.$$

To account for noise in the measurement data, we need to replace the consistency loss \mathcal{L}_{MC} by one of the self-supervised losses for noisy data presented in Chapter 2, with the specific choice depending on the amount of knowledge we have about the noise distribution:

a) If the measurements have noise with a known distribution (e.g., Poisson, Gaussian, etc.), we can use the (GR2R) or (SURE) loss, i.e.,

$$\begin{aligned} \mathcal{L}_{\text{GR2R-MSPLIT}}(\mathbf{y}, \mathbf{A}, f) &= \\ \mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2, \mathbf{A}_1, \mathbf{A}_2 | \mathbf{y}, \mathbf{A}} \left\{ \mathcal{L}_{\text{GR2R}}(\mathbf{y}_1, \mathbf{A}_1 \circ f) + \frac{1}{m} \|\mathbf{A}_2 f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{y}_2\|^2 \right\} \end{aligned}$$

As we saw in Section 2.5, the first term is an unbiased estimator of the noiseless measurement consistency, i.e.,

$$\mathbb{E}_{\mathbf{y}_1, \mathbf{A}_1 | \mathbf{x}} \{\mathcal{L}_{\text{GR2R}}(\mathbf{y}_1, \mathbf{A}_1 \circ f)\} = \mathbb{E}_{\mathbf{y}_1, \mathbf{A}_1 | \mathbf{x}} \left\{ \frac{1}{m} \|\mathbf{A}_1 f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{A}_1 \mathbf{x}\|^2 \right\}$$

and thus use a similar analysis to that in Proposition 3.2, to conclude that the minimizer of this loss approximates $\mathbb{E}_{\mathbf{x}|\mathbf{y}_1, \mathbf{A}_1}\{\mathbf{x}\}$ if $\mathbf{Q}_{\mathbf{A}_1}$ is invertible and a single split is used at test time, or $\mathbb{E}_{\mathbf{A}_1|\mathbf{A}}\left\{\bar{\mathbf{Q}}_{\mathbf{A}}^{-1}\mathbf{Q}_{\mathbf{A}_1}\mathbb{E}_{\mathbf{x}|\mathbf{y}_1, \mathbf{A}_1}\{\mathbf{x}\}\right\}$ if $\bar{\mathbf{Q}}_{\mathbf{A}}$ is invertible and multiple splits are used at test time.

b) If the measurements have an unknown noise distribution which can be assumed to be separable across measurements, we can simply remove the measurement consistency loss, such that the resulting loss, known by the name SSDU [27], can be seen as a multioperator extension of (SPLIT):

$$\mathcal{L}_{\text{SSDU}}(\mathbf{y}, \mathbf{A}, f) = \mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2, \mathbf{A}_1, \mathbf{A}_2 | \mathbf{y}, \mathbf{A}} \left\{ \frac{1}{m} \|\mathbf{A}_2 f(\mathbf{y}_1, \mathbf{A}_1) - \mathbf{y}_2\|^2 \right\}$$

In this case, we can derive a similar result than that in Proposition 3.2, but where $\mathbf{Q}_{\mathbf{A}_1} = \mathbb{E}_{\mathbf{A}_2|\mathbf{A}_1}\{\mathbf{A}_2^\top \mathbf{A}_2\}$ instead of $\mathbb{E}_{\mathbf{A}|\mathbf{A}_1}\{\mathbf{A}^\top \mathbf{A}\}$. As \mathbf{A}_2 and \mathbf{A}_1 do not overlap, $\mathbf{Q}_{\mathbf{A}_1}$ does not cover the range of \mathbf{A}_1^\top , and $\mathbf{Q}_{\mathbf{A}_1}$ will not be invertible for any split. Nonetheless, we can average over multiple splits at test time as in (3.13), such that $\bar{\mathbf{Q}}_{\mathbf{A}} = \frac{1}{J} \sum_{j=1}^J \mathbf{Q}_{\mathbf{A}_1^{(j)}}$ becomes invertible, and the test time estimator, f^{test} , approximates the conditional estimator $\mathbb{E}_{\mathbf{A}_1|\mathbf{A}}\left\{\bar{\mathbf{Q}}_{\mathbf{A}}^{-1}\mathbf{Q}_{\mathbf{A}_1}\mathbb{E}_{\mathbf{x}|\mathbf{y}_1, \mathbf{A}_1}\{\mathbf{x}\}\right\}$.

Consistency across operators The main drawback of splitting-based approaches is that they arguably do not input all the measurement information available to the network. The multi-operator imaging (MOI) loss [87] addresses this issue by assuming that each of the imaging problems is invertible, i.e., there exists an f^* such that $\mathbf{x} \approx f^*(\mathbf{Ax}, \mathbf{A})$ for each \mathbf{A} , and then enforcing estimator consistency across the different operators. That is $f(\mathbf{Ax}, \mathbf{A}) \approx f(\mathbf{A}'\mathbf{x}, \mathbf{A}')$ for any pair of operators $\mathbf{A} \neq \mathbf{A}'$ belonging to $p(\mathbf{A})$:

$$\mathcal{L}_{\text{MOI}}(\mathbf{y}, \mathbf{A}, f) = \mathbb{E}_{\mathbf{A}' \sim p(\mathbf{A})} \left\{ \frac{1}{n} \|f(\mathbf{A}'f(\mathbf{y}, \mathbf{A}), \mathbf{A}') - f(\mathbf{y}, \mathbf{A})\|^2 \right\} \quad (\text{MOI})$$

The loss is minimized together with measurement consistency, leading to the following total loss:

$$\mathcal{L}(\mathbf{y}, \mathbf{A}, f) = \mathcal{L}_{\text{MC}}(\mathbf{y}, \mathbf{A} \circ f) + \lambda \mathcal{L}_{\text{MOI}}(\mathbf{y}, \mathbf{A}, f) \quad (3.14)$$

where $\lambda > 0$ is a trade-off hyperparameter. It is easy to verify that the trivial reconstruction $f(\mathbf{y}, \mathbf{A}) = \mathbf{A}^\dagger \mathbf{y}$ is not a minimizer of this loss, as long as $\mathbf{A}'\mathbf{A}^\dagger \neq \mathbf{A}\mathbf{A}'^\dagger$ for some $\mathbf{A} \neq \mathbf{A}'$. In the case of noisy measurements, the \mathcal{L}_{MC} can be replaced by any of the losses introduced in Chapter 2 according to the amount of knowledge we have about the noise distribution, in a similar way as the robust extensions of (MSPLIT).

While the MOI loss does not have the nice equivalence to the supervised loss that we saw in with the splitting losses, it is able to leverage all the available measurements and can be theoretically motivated by the model identifiability theory discussed in Section 3.4.

3.2 Leveraging invariance to transformations

In applications where we have a single non-invertible forward operator, Chen et al. [26] show that it still possible learn in its nullspace if we assume that the distribution of signals is invariant to a group of transformations $\mathbf{T}_g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ for $g = 1, \dots, G$, such as rotations or

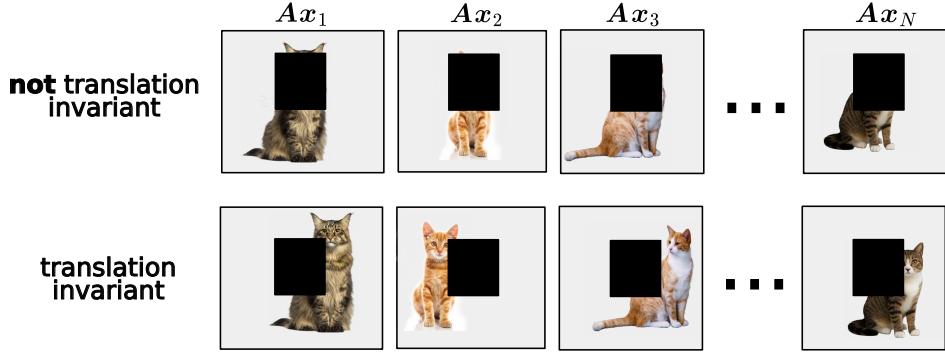


Figure 3.3: **Example of the equivariant imaging principle.** Consider a toy setting where we image cats through a fixed masking operator. If the cat distribution is not invariant to translations (top row), the cats have a canonical position in the image, and we never observe a part of the distribution (the heads). If the distribution is invariant (bottom row), the cats are not always centered, and we can learn the cat distribution (including their heads) despite never observing the masked pixels.

translations acting on a discretized image of n pixels. Mathematically speaking, if the support of the distribution is invariant, we have that for every transform \mathbf{T}_g , if $\mathbf{x} \in \text{supp}(p_{\mathbf{x}})$, then $\mathbf{T}_g(\mathbf{x}) \in \text{supp}(p_{\mathbf{x}})$. This condition is less strict than asking the distribution to be invariant, i.e., $p(\mathbf{T}_g(\mathbf{x})) = p(\mathbf{x})$ for all transformations \mathbf{T}_g and signals \mathbf{x} . Due to the invariance property, we have the following key observation:

$$\mathbf{y} = \mathbf{A}\mathbf{x} = \mathbf{A} \circ \mathbf{T}_g \circ \mathbf{T}_g^{-1}(\mathbf{x}) = \mathbf{A} \circ \mathbf{T}_g(\mathbf{x}') \quad (3.15)$$

where $\mathbf{x}' = \mathbf{T}_g^{-1}(\mathbf{x})$ also belongs to the signal set. Thus, the invariance property *implicitly* provides us with additional virtual observations through a family of different operators $\{\mathbf{A}_g = \mathbf{A} \circ \mathbf{T}_g\}_{g=1}^G$ and we are in a similar but slightly more constrained setup to that in Section 3.1 (see Figure 3.3 for an illustration of this idea). We will see that this enables us to exploit a powerful property called equivariance.

Equivariance The concept of equivariance has been widely studied in machine learning, especially in the context of incorporating symmetries into neural network architectures [88]. Informally, a function is said to be equivariant to a group of transformations if applying a transformation to the input results in a corresponding transformation of the output. More formally, we have the following definition:

Definition 3.4 (Equivariance [89]). *A function $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is equivariant to a group action if: $\tilde{\mathbf{T}}_g(\phi(\mathbf{x})) = \phi(\mathbf{T}_g(\mathbf{x}))$ for all $g = 1, \dots, G$, where $\mathbf{T}_g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $\tilde{\mathbf{T}}_g : \mathbb{R}^m \rightarrow \mathbb{R}^m$ are (possibly the same if $n = m$) transformations satisfying the group properties.*

In the context of inverse problems, we cannot directly apply this definition to the reconstruction function f due to its delicate interplay with the forward operator. We can use instead a more specific definition that takes into account this interplay:

Definition 3.5 (Equivariant reconstructor [90]). *We say that the $f(\mathbf{y}, \mathbf{A})$ is an equivariant reconstructor if*

$$f(\mathbf{y}, \mathbf{AT}_g) = \mathbf{T}_g^{-1} f(\mathbf{y}, \mathbf{A}), \quad \forall \mathbf{y} \in \mathbb{R}^m, \forall g, \forall \mathbf{A} \in \mathbb{R}^{m \times n}. \quad (3.16)$$

Below we illustrate how popular reconstructors are equivariant as long as their building blocks are equivariant in the sense of Definition 3.4 or if the signal distribution $p_{\mathbf{x}}$ is invariant. Proofs can be found in [90].

1. **Back-projection networks.** The backprojection network $f(\mathbf{y}, \mathbf{A}) = \phi(\mathbf{A}^\dagger \mathbf{y})$ is an equivariant reconstructor if the image-to-image mapping ϕ is equivariant.
2. **Unrolled network.** The unrolled network defined in (1.6) is an equivariant reconstructor for any stepsize $\tau \in \mathbb{R}$ if the proximal operators ϕ_1, \dots, ϕ_k are equivariant.
3. **Reynolds averaging.** Let $\tilde{f}(\mathbf{y}, \mathbf{A})$, be any (non-equivariant) reconstructor, then

$$f(\mathbf{y}, \mathbf{A}) = \frac{1}{G} \sum_{g=1}^G \mathbf{T}_g \tilde{f}(\mathbf{y}, \mathbf{AT}_g) \quad (3.17)$$

is an equivariant reconstructor.

4. **Variational methods.** The variational estimate

$$f(\mathbf{y}, \mathbf{A}) = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}(\mathbf{x})\|^2 + \rho(\mathbf{x}). \quad (3.18)$$

is an equivariant reconstructor if the regularization term ρ is an invariant distribution. In particular, for observations with isotropic Gaussian noise and $\rho(\mathbf{x}) = -\log p_{\mathbf{x}}(\mathbf{x})$, then f corresponds to the maximum-a-posteriori estimate, and is equivariant if $p_{\mathbf{x}}$ is an invariant distribution.

5. **MMSE.** The MMSE estimate

$$f(\mathbf{y}, \mathbf{A}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}, \mathbf{A}} \{\mathbf{x}\} \quad (3.19)$$

is an equivariant reconstructor if $p_{\mathbf{x}}$ is an invariant distribution.

The first three examples show how to build practical equivariant reconstructors, while the last two examples show that optimal Bayesian reconstructors are equivariant when the signal distribution is invariant.

At this point, one might ask whether the equivariant reconstructor property in Definition 3.5, together with a simple measurement consistency loss (MC) is sufficient for learning with a single non-invertible operator \mathbf{A} . The answer is negative, since the simple linear pseudo-inverse $f(\mathbf{y}, \mathbf{A}) = \mathbf{A}^\dagger \mathbf{y}$ is an equivariant reconstructor that minimizes the measurement consistency loss, but does not recover any information in the nullspace of \mathbf{A} . As with the multi-operator case, we can use a splitting loss or enforce consistency across transforms to further constrain the solution space.

Equivariant splitting loss We can follow a similar approach to the (MSPLIT) loss in the previous section, but this time using operators related by a transformation, resulting in the equivariant splitting loss [90]:

$$\begin{aligned}\mathcal{L}_{\text{ESPLIT}}(\mathbf{y}, f) &= \frac{1}{G} \sum_{g=1}^G \mathcal{L}_{\text{MSPLIT}}(\mathbf{y}, \mathbf{A}\mathbf{T}_g, f) \\ &= \frac{1}{G} \sum_{g=1}^G \mathbb{E}_{\mathbf{y}_1, \mathbf{A}_1 | \mathbf{y}, \mathbf{A}} \{ \| \mathbf{A}\mathbf{T}_g f(\mathbf{y}_1, \mathbf{A}_1 \mathbf{T}_g) - \mathbf{y} \|^2 \}\end{aligned}\quad (\text{ESPLIT})$$

where $\mathbf{A}_1 \sim p(\mathbf{A}_1 | \mathbf{A})$ is a random split of \mathbf{A} . As with Proposition 3.2 in the multi-operator splitting case, minimizing this loss under the assumption of an invariant signal distribution $p_{\mathbf{x}}$ can recover the MMSE estimator, as long as $\mathbf{Q}_{\mathbf{A}_1} = \sum_{g \in \mathcal{S}_{\mathbf{A}_1}} \mathbf{T}_g^\top \mathbf{A}^\top \mathbf{A} \mathbf{T}_g$, where $\mathcal{S}_{\mathbf{A}_1}$ is comprised by all transforms for which \mathbf{A}_1 may arise as a random split of $\mathbf{A}\mathbf{T}_g$, or $\bar{\mathbf{Q}}_{\mathbf{A}} = \mathbb{E}_{\mathbf{A}_1 | \mathbf{A}} \{ \mathbf{Q}_{\mathbf{A}_1} \}$ are full rank. Since summing over the whole group of transformations can be expensive, the loss can be evaluated by randomly sampling a transformation at each training iteration. As with the multi-operator splitting loss, the test time estimator can be computed by averaging over multiple splits and transformations as in (3.13), and a noise-robust extension of the loss can be derived in a similar way by replacing the term enforcing consistency with the input \mathbf{y}_1 by (GR2R) or (SURE).

If we choose f to be an equivariant reconstructor by design, for example using a back-projection network with an equivariant denoiser architecture, the equivariant splitting loss in (ESPLIT) reduces to the simpler splitting loss in (MSPLIT) with a single operator, as stated in the following proposition:

Proposition 3.6. *If f is an equivariant reconstructor, then (ESPLIT) is equivalent to the splitting loss*

$$\mathcal{L}_{\text{ESPLIT}}(\mathbf{y}, f) = \mathcal{L}_{\text{MSPLIT}}(\mathbf{y}, \mathbf{A}, f). \quad (3.20)$$

The proof follows directly from the definition of equivariant reconstructor in Definition 3.5. This result shows how training on a simple splitting loss with a single operator can be effective even in the case of incomplete measurements, as long as the network is an equivariant reconstructor by construction.

Consistency across transforms Following the same logic behind (MOI), if we assume that the imaging problem is approximately invertible, i.e., there exists a function, f^* , such that $\mathbf{x} \approx f^*(\mathbf{A}\mathbf{x}, \mathbf{A})$ for all $\mathbf{x} \in \text{supp}(p_{\mathbf{x}})$, then, as illustrated in Figure 3.4, a good reconstruction function f should be such that the *full imaging/sensing system*, $f \circ \mathbf{A}$ is approximately equivariant

$$f(\mathbf{A}\mathbf{T}_g(\mathbf{x}), \mathbf{A}) \approx \mathbf{T}_g f(\mathbf{A}\mathbf{x}, \mathbf{A}). \quad (3.21)$$

The equivariant imaging (EI) loss aims at enforcing the system equivariance via training [26]:

$$\mathcal{L}_{\text{EI}}(\mathbf{y}, f) = \frac{1}{G} \sum_{g=1}^G \frac{1}{n} \| f(\mathbf{A}\mathbf{T}_g f(\mathbf{y}, \mathbf{A}), \mathbf{A}) - \mathbf{T}_g f(\mathbf{y}, \mathbf{A}) \|^2. \quad (\text{EI})$$

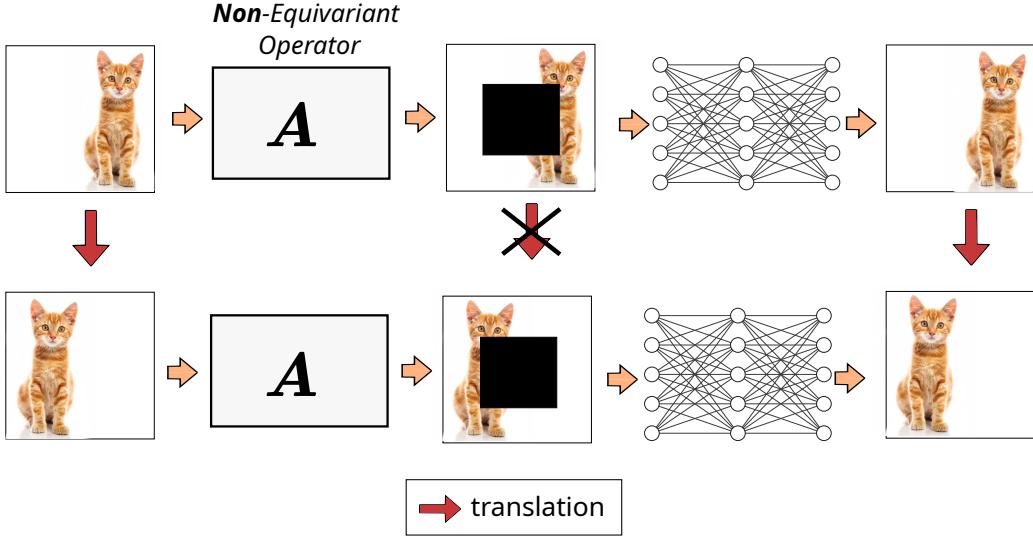


Figure 3.4: The equivariant imaging loss attempts to enforce equivariance of the imaging system to a group of transformations (here translations). This loss can enable learning in the nullspace of the operator (here missing pixels), as the transformations implicitly *translate* the nullspace.

The equivariant system condition is stronger than asking f to be an equivariant reconstructor as in Definition 3.5, and in some cases will only hold exactly if the imaging problem is invertible². Thus, the EI loss is necessary even when using equivariant reconstructors, as it further constrains the solution space.

As with the MOI loss, this loss should be minimized together with measurement consistency:

$$\mathcal{L}(\mathbf{y}, f) = \mathcal{L}_{\text{MC}}(\mathbf{y}, f) + \lambda \mathcal{L}_{\text{EI}}(\mathbf{y}, f) \quad (3.22)$$

where $\lambda > 0$ is a hyperparameter and the \mathcal{L}_{MC} can be replaced in the case of noisy measurements by any of the losses introduced in Chapter 2, such as (SURE) [62].

Conditions on \mathbf{A} with linear transformations Typical transforms such as rotations or translations are linear operators, and we can think of them as a collection of invertible matrices $\{\mathbf{T}_g\}_{g=1}^G$. In this case, we can follow a similar analysis to the multiple operator setting in the previous section with operators defined as $\mathbf{A}_g = \mathbf{A}\mathbf{T}_g$. As we saw in Section 3.1, we need that $\mathbf{A}\mathbf{T}_g$ have different nullspaces, or in other words, that $\sum_g \mathbf{T}_g^\top \mathbf{A}^\top \mathbf{A}\mathbf{T}_g$ is an invertible matrix. An important consequence is that the forward operator should *not* be equivariant to the group of transformations:

Proposition 3.7 (Tachella et al. [79]). *If $\mathbf{A}^\top \mathbf{A}$ is equivariant to the group of transformations $\{\mathbf{T}_g\}_{g=1}^G$, then all operators $\mathbf{A}\mathbf{T}_g$ share the same nullspace.*

Equivariance means that $\mathbf{A}^\top \mathbf{A}\mathbf{T}_g = \mathbf{T}_g \mathbf{A}^\top \mathbf{A}$ for all $g = 1, \dots, G$, and therefore that \mathbf{A} shares the same nullspace with $\mathbf{A}\mathbf{T}_g$ for all g .

²For example, the MMSE estimate under an invariant distribution $p_{\mathbf{x}}$ is always an equivariant reconstructor, but it only satisfies this stronger condition if the imaging problem is invertible, i.e., $\mathbb{E}_{\mathbf{x}|\mathbf{y}, \mathbf{A}} \{\mathbf{x}\} = \mathbf{x}$.

As illustrated in the following examples, the requirement for \mathbf{A} not being equivariant cannot be taken for granted, and in any given scenario the choice of group actions will depend on specific properties of the forward operator:

- Compressive random operators $\mathbf{A} \sim \mathcal{N}(\mathbf{0}, \frac{1}{m}\mathbf{I})$ with $m < n$ are not equivariant to any (non-trivial) set of transformations $\{\mathbf{T}_g\}_{g=1}^G$, except for the amplitude scalings $\mathbf{T}_g = g\mathbf{I}$, with probability 1.
- Image inpainting $\mathbf{A} = \text{diag}(\mathbf{b})$ is generally not equivariant to pixel shifts, as missing pixels have fixed locations in the image.
- Operators that admit a diagonalization $\mathbf{A} = \mathbf{Q} \text{diag}(\mathbf{b}) \mathbf{F}$ where \mathbf{F} is the Fourier transform and \mathbf{Q} is any unitary basis, such as any blurring operation or MRI, are equivariant to pixel shifts or translations. However, if the blurs have a specific orientation or the MRI masks are accelerated using a Cartesian subsampling pattern, these operators are not equivariant to rotations.
- Isotropic blurs and downsampling with antialiasing filters are equivariant to both rotations and translations. However, they are not equivariant to scaling transformations which can be used to learn to reconstruct the missing high-frequencies [91].

While the EI loss does not carry with it a strong equivalence with a supervised loss, it shares the same motivation as MOI from the model identifiability theory discussed in Section 3.4.

3.3 Learning generative models from incomplete measurements

So far, we have presented multiple losses than can approximate the posterior mean of the problem. Here we go further and ask whether we can identify the signal distribution, $p_{\mathbf{x}}$, from incomplete measurement data alone. In this section, we present the main approaches that have been explored so far, including generative adversarial networks (GANs) and diffusion models.

Generative adversarial networks In the unconditional generation setting, we aim to train a generator $f : \mathbb{R}^k \mapsto \mathbb{R}^n$ mapping latents $\mathbf{z} \in \mathbb{R}^k$ following a simple distribution such as an isotropic Gaussian, to samples $\mathbf{x} \in \mathbb{R}^n$ of the image distribution $p_{\mathbf{x}}$. We can aim to match distribution of measurements associated with the generated distribution $p_{\hat{\mathbf{y}}}$ defined as

$$\begin{cases} \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \mathbf{A} \sim p(\mathbf{A}) \\ \hat{\mathbf{x}} = f(\mathbf{z}) \\ \hat{\mathbf{y}} \sim p(\mathbf{y} | \mathbf{A} \hat{\mathbf{x}}) \end{cases} \quad (3.23)$$

to the distribution of observed measurements $p_{\mathbf{y}}$. AmbientGAN [82] proposes to train a Wasserstein GAN [92] to achieve this goal:

$$\min_f \max_d \mathbb{E}_{\mathbf{z}, \mathbf{A}} \{d(\mathbf{A}f(\mathbf{z}), \mathbf{A})\} - \mathbb{E}_{\mathbf{y}, \mathbf{A}} \{d(\mathbf{y}, \mathbf{A})\} \quad (3.24)$$

where $d : \mathbb{R}^n \times \mathbb{R}^{m \times n} \mapsto \mathbb{R}$ is a 1-Lipschitz discriminator network trained jointly with the generator, which can incorporate information about the forward operator. The first term in (3.24) pushes f to generate realistic measurements by fooling the discriminator, whereas the second term trains d to discriminate real measurements from generated ones.

This idea can be also extended to a conditional GAN model [93, 94], where the generator, $f(\mathbf{z}, \mathbf{y}, \mathbf{A})$, is conditioned on a measurement and operator, and aims to generate posterior samples with varying latents \mathbf{z} .

Diffusion models AmbientDiffusion [85] extends the AmbientGAN idea to diffusion models, in the case of *noiseless* but incomplete measurements from multiple forward operators. In this setting, a reconstruction network is trained using the (MSPLIT) loss at different noise levels by adding synthetic Gaussian noise with standard deviation, σ , to the input measurements \mathbf{y}_1 , thus approximating the conditional estimator $f(\mathbf{y}_1, \mathbf{A}_1, \sigma) = \mathbb{E}_{\mathbf{x}|\mathbf{y}_1+\sigma\epsilon, \mathbf{A}_1} \{\mathbf{x}\}$. Once the network is trained, samples of the signal distribution [85] or posterior [95] are obtained by fixing a random split \mathbf{A}_1 , and using f as a proxy for the Gaussian denoiser $\mathbb{E}_{\mathbf{x}|\mathbf{x}+\sigma\epsilon} \{\mathbf{x}\}$ required to run the diffusion.

In a similar multioperator setting, Rozet el al. [43] propose a different approach based on expectation-maximization, which consists of iterating between i) generating posterior samples using a diffusion approach with a fixed denoiser network, and then ii) updating the denoiser on the generated samples. At initialization, the denoiser is initialized to sample from a Gaussian distribution.

3.4 Model identification theory

While we have seen that splitting losses can approximate the supervised ℓ_2 loss under certain assumptions, it is important to ask if the harder task of learning a generative model is even well defined, or in other words, if we can uniquely identify $p_{\mathbf{x}}$ from incomplete measurements³, without any additional assumptions on $p_{\mathbf{x}}$? The question can be answered by analyzing the available information about the characteristic function of the signal distribution, $\phi_{\mathbf{x}}(\boldsymbol{\omega})$. For each operator $\mathbf{A} \sim p(\mathbf{A})$, we have

$$\phi_{\mathbf{y}|\mathbf{A}}(\tilde{\boldsymbol{\omega}}) = \mathbb{E}_{\mathbf{y}|\mathbf{A}} \left\{ e^{i\tilde{\boldsymbol{\omega}}^\top \mathbf{y}} \right\} = \mathbb{E}_{\mathbf{x}} \left\{ e^{i\tilde{\boldsymbol{\omega}}^\top \mathbf{A}\mathbf{x}} \right\} = \mathbb{E}_{\mathbf{x}} \left\{ e^{i(\mathbf{A}^\top \tilde{\boldsymbol{\omega}})^\top \mathbf{x}} \right\} \quad (3.25)$$

$$= \phi_{\mathbf{x}}(\boldsymbol{\omega} = \mathbf{A}^\top \tilde{\boldsymbol{\omega}}) \text{ with } \tilde{\boldsymbol{\omega}} \in \mathbb{R}^m \quad (3.26)$$

Thus, for each operator we observe the characteristic function of \mathbf{x} in the range of \mathbf{A}^\top , which is an m -dimensional linear subspace of \mathbb{R}^n as $m < n$. Since a distribution is uniquely determined by its characteristic function, we need to observe infinitely-many operators in order to fully cover the characteristic function of \mathbf{x} ! This observation dates back to the work by Cramér and Wold, which focuses in the case $m = 1$:

Theorem 3.8 (Cramér and Wold [96]). *A probability distribution $p(\mathbf{x})$ is uniquely determined by the totality of its one-dimensional projections.*

³We do not consider noise in this part, since we can take noise into account by reasoning in two steps: first we apply the identification results from noisy data in Proposition 2.9 to first identify the clean measurement distribution from the noisy one for each forward operator, and second, identify the signal distribution from the set of clean measurement distributions [79].

The theorem says that if we observe (unpaired) scalar measurements $y = \mathbf{a}^\top \mathbf{x} \in \mathbb{R}$ with $\mathbf{x} \sim p_{\mathbf{x}}(\mathbf{x})$ and a measurement distribution $\mathbf{a} \sim p(\mathbf{a})$ that covers the whole space of possible projections (i.e., is dense in \mathbb{R}^n), we can uniquely identify the $p_{\mathbf{x}}$. Unfortunately, this result is not very practical, since it only holds in the limit of observing *all possible* projections in \mathbb{R}^n , whereas we generally only obtain observations via a finite number of operators, and thus the distribution is not dense in \mathbb{R}^n (e.g., is limited to varying masks in image inpainting or accelerated MRI).

3.4.1 Identification of low-dimensional distributions

In order to obtain sufficient conditions in the more realistic setting of a finite number of operators, we need to consider some additional assumptions on the signal distribution, $p_{\mathbf{x}}$. In the following, we will see that assuming that the signal distribution is low-dimensional, or in other words, that the support of $p_{\mathbf{x}}$ is a low-dimensional subset of \mathbb{R}^n , is sufficient for obtaining model identification guarantees. Low-dimensionality is a common assumption in imaging and data science, and it is often referred to as the *manifold hypothesis* [97].

Example 3.9. We can illustrate why and how low-dimensionality can help by considering a simple example where $p_{\mathbf{x}}$ is supported on a k -dimensional subspace of dimension k , such that we can write any signal as $\mathbf{x} = \mathbf{Q}\mathbf{z}$ for some low-dimensional latent vector $\mathbf{z} \sim p_{\mathbf{z}}$ and a fixed linear decoder $\mathbf{Q} \in \mathbb{R}^{n \times k}$. Assuming that the linear decoder \mathbf{Q} is known, we can observe the characteristic function of the latent variable as $\phi_{\mathbf{y}|\mathbf{A}}(\tilde{\omega}) = \phi_{\mathbf{z}}(\omega = (\mathbf{AQ})^\top \tilde{\omega})$ for all $\tilde{\omega} \in \mathbb{R}^m$ and $g = 1, \dots, G$. If we further assume that $m \geq k$ and $\text{rank}(\mathbf{AQ}) = k$ for some \mathbf{A} , we can follow a similar argument to that in (3.25) to conclude that we can uniquely identify the latent distribution $p_{\mathbf{z}}$, and thus also uniquely identify $p_{\mathbf{x}}$ as the linear decoder is known.

The intuition of the linear subspace can be generalized to more general k -dimensional sets. In the example, the two key steps for model identification are to i) identify the low-dimensional support of the distribution, which we denote as $\text{supp}(p_{\mathbf{x}}) := \mathcal{X}$ (in the linear case, the support is given by the range of the linear decoder \mathbf{Q}), and ii) require that \mathbf{A} is one-to-one when restricted to \mathcal{X} (in the linear case, this is equivalent to asking $\text{rank}(\mathbf{AQ}) = k$).

In the noiseless measurements case, if there is a one-to-one reconstruction map f between the measurement set, $\text{supp}(p_{\mathbf{y}}) = \mathcal{Y}$, and the signal set, \mathcal{X} , then we can identify the signal distribution $p_{\mathbf{x}}$ by simply applying f to all measurements in \mathcal{Y} . We thus focus on the problem of identifying the support \mathcal{X} from the measurement distribution one.

While multiple definitions of low-dimensionality of a set exist [98], here we focus on the upper box-counting dimension⁴, which is convenient for the theoretical results, and covers both well behaved models such as compact manifolds where the definition coincides with the more intuitive topological dimension, as well as more general sets. The following theorem provides a sufficient condition for uniquely identifying the signal set from measurements associated to multiple forward operators (c.f., Section 3.1):

⁴The box-counting dimension [98, Chapter 2] is defined for a compact subset $\mathcal{S} \subset \mathbb{R}^n$ as

$$\text{boxdim}(\mathcal{S}) = \limsup_{\epsilon \rightarrow 0} \frac{\log \mathbb{N}(\mathcal{S}, \epsilon)}{-\log \epsilon} \quad (3.27)$$

where $\mathbb{N}(\mathcal{S}, \epsilon)$ is the minimum number of closed balls of radius ϵ with respect to the norm $\|\cdot\|$ that are required to cover \mathcal{S} .

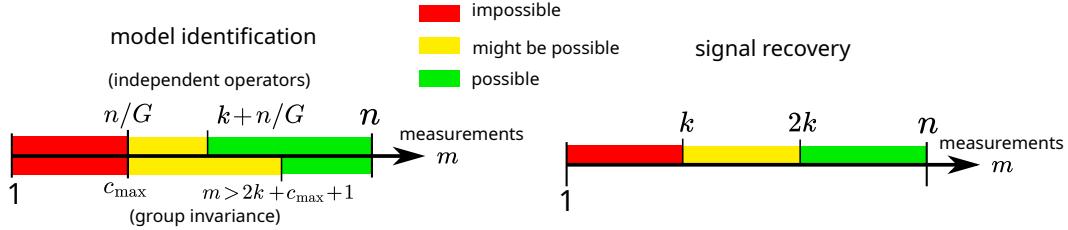


Figure 3.5: Model identification and signal recovery regimes [79] as a function of the number of partial observations m per signal, model dimension k , ambient dimension n and number of measurement operators G (when it is possible to access multiple independent operators) or maximum multiplicity of an invariant subspace c_{\max} (when the signal is group invariant or the operators are related via a group action).

Theorem 3.10 (Tachella et al. [79]). *Assume that the signal set \mathcal{X} is a bounded set with box-counting dimension k . For almost every set of G operators $\mathbf{A}_1, \dots, \mathbf{A}_G \in \mathbb{R}^{m \times n}$, the signal model \mathcal{X} can be uniquely identified from the measurement sets $\{\mathbf{A}_g \mathcal{X}\}_{g=1}^G$ if the number of measurements per operator verifies $m > k + n/G$.*

It is worth noting that this result does not directly apply for *any* set of G operators (e.g., specific MRI or inpainting operators), but rather requires G generic operators, which removes degenerate cases. Nonetheless, it provides us with fundamental bounds on the number of measurements and operators needed to uniquely identify the signal distribution.

We can further refine this result for the more constrained case where the operators are linked by a group of transformations (c.f., Section 3.2), i.e., $\{\mathbf{A}_g = \mathbf{AT}_g\}_{g=1}^G$. The following theorem relies on the dimension of the largest linear subspace of \mathbb{R}^n which is invariant to the group of transformations⁵, which we denote by c_{\max} .

Theorem 3.11 (Tachella et al. [79]). *Let $\{\mathbf{T}_g\}_{g=1}^G$ be a group of transformations associated with a compact cyclic group and assume that the signal set \mathcal{X} is a bounded set with box-counting dimension k . For almost every $\mathbf{A} \in \mathbb{R}^{m \times n}$, the signal set can be uniquely identified from the sets $\{\mathbf{AT}_g \mathcal{X}\}_{g=1}^G$ if the number of measurements verifies $m > 2k + c_{\max} + 1$.*

Using the fact that for any compact cyclic group $c_{\max} \geq n/G$, when the equality holds we get the bound $m > 2k + n/G + 1$ which resembles that of Theorem 3.10, although requiring $2k$ measurements instead of k . It turns out that these additional measurements are necessary in some cases, since it is possible to build a counter-example where identifying the support of $p_{\mathcal{X}}$ is impossible using any operator with $m \leq 2k + c_{\max} - 2$ measurements [79].

Comparison with signal recovery theory A well-known result from compressed sensing [99] and embedding theory [100], is that a sufficient condition for a generic $\mathbf{A} \in \mathbb{R}^{m \times n}$ to be one-to-one on the support of the signal distribution \mathcal{X} is to have a number of measurements larger than two times the dimension of the set, that is $m > 2k$ where k is the box-counting dimension of \mathcal{X} . These results are known as signal recovery theorems, since they specify the minimum number of measurements that guarantee the existence of a reconstruction function

⁵The maximum multiplicity of the group action c_{\max} , is given by the largest dimension of a linear subspace $\mathcal{S} \subseteq \mathbb{R}^n$ such that $\mathbf{T}_g \mathcal{S} \subseteq \mathcal{S}$ for all $g = 1, \dots, G$. See [79] for more details.

perfectly recovering all plausible signals. Figure 3.5 compares the necessary and sufficient conditions for model identification presented in this chapter with those for unique signal recovery.

Relationship to splitting Building on Proposition 3.2, we can also ask what are the minimum numbers of operators and measurements in order to uniquely identify the signal distribution via splitting losses and how these compare to the results in Theorem 3.10.

The best-case scenario of G operators that verify the conditions in Proposition 3.2 can be constructed in the following way: consider operators given by $\mathbf{A} = [\mathbf{A}_1^\top, \mathbf{A}_2^\top]^\top \in \mathbb{R}^{m \times n}$ where $\mathbf{A}_1 \in \mathbb{R}^{(2k+1) \times n}$ is fixed, and $\mathbf{A}_2 \in \mathbb{R}^{(m-2k+1) \times n}$ is sampled as one out of G operators, i.e., $\mathbf{A}_2 \sim \frac{1}{G} \sum_{g=1}^G \delta_{\mathbf{A}_{2,g}}$, independently of \mathbf{A}_1 . In order to learn the conditional estimator $\mathbb{E}_{\mathbf{x}|\mathbf{y}_1, \mathbf{A}_1} \{\mathbf{x}\}$, Proposition 3.2 requires that $\mathbf{Q}_{\mathbf{A}_1} = \mathbb{E}_{\mathbf{A}|\mathbf{A}_1} \{\mathbf{A}^\top \mathbf{A}\} \in \mathbb{R}^{n \times n}$ is invertible (and thus has full rank). Assuming that we are in the noiseless setting and that the signal distribution has dimension $k \leq n$, we can choose $2k+1$ measurements in \mathbf{A}_1 , such that we have unique signal recovery [100]. Thus, the conditional estimator obtains a perfect reconstruction, $\mathbb{E}_{\mathbf{x}|\mathbf{y}_1, \mathbf{A}_2} \{\mathbf{x}\} = \mathbf{x}$. We can then uniquely identify $p_{\mathbf{x}}$ by simply reconstructing measurements from $p_{\mathbf{y}}$ once we have learned the conditional mean estimator. Due to the simplified form of \mathbf{A} , we can compute $\mathbf{Q}_{\mathbf{A}_1}$ of Proposition 3.2 in closed form as

$$\begin{aligned} \mathbf{Q}_{\mathbf{A}_1} &= \mathbb{E}_{\mathbf{A}|\mathbf{A}_1} \{\mathbf{A}^\top \mathbf{A}\} \\ &= \mathbb{E}_{\mathbf{A}_2} \{\mathbf{A}_2^\top \mathbf{A}_2\} + \mathbb{E}_{\mathbf{A}|\mathbf{A}_1} \{\mathbf{A}_1^\top \mathbf{A}_1\} \\ &= \frac{1}{G} \sum_{g=1}^G \mathbf{A}_{2,g}^\top \mathbf{A}_{2,g} + \mathbf{A}_1^\top \mathbf{A}_1. \end{aligned}$$

Since $\mathbf{Q}_{\mathbf{A}_1}$ is composed of the sum of G matrices of rank at most $m - (2k+1)$ and one matrix of rank at most $2k+1$, we have that $\text{rank}(\mathbf{Q}_{\mathbf{A}_1}) \leq \min\{G(m-2k-1) + 2k+1, n\}$ with equality for a generic choice of $\mathbf{A}_{2,g}$.

As we need $\text{rank}(\mathbf{Q}_{\mathbf{A}_1}) = n$, it is necessary that $G(m-2k-1) + 2k+1 \geq n$. Thus we obtain the condition

$$m \geq \frac{n}{G} + \left(1 - \frac{1}{G}\right)(2k+1)$$

which is a similar, albeit less tight, version of the sufficient condition in Theorem 3.10.

3.5 Summary

Self-supervised learning in inverse problems where the forward operator is many-to-one is possible as long as the operators change across samples, or if we can assume that the signal distribution is invariant to a group of transformations, such as translations, rotations or scalings.

Table 3.1 summarizes the assumptions behind the different families of self-supervised losses introduced in this chapter. The assumptions focus on the conditions on the forward operators rather than the noise, as all losses can be adapted to handle noise following the principles introduced in Chapter 2.

Family	Assumptions	Refs.
(MSPLIT)	Necessary Multiple forward operators $\mathbb{E}_{\mathbf{A}} \{\mathbf{A}^\top \mathbf{A}\}$ is invertible <hr/> Sufficient (cf. Proposition 3.2) $\mathbb{E}_{\mathbf{A}_1 \mathbf{A}} \{\mathbb{E}_{\mathbf{A} \mathbf{A}_1} \{\mathbf{A}^\top \mathbf{A}\}\}$ is invertible	[27, 85]
(MOI)	Necessary Multiple forward operators $\mathbb{E}_{\mathbf{A}} \{\mathbf{A}^\top \mathbf{A}\}$ is invertible $\exists f^*$ such that $\mathbf{x} \approx f^*(\mathbf{Ax}, \mathbf{A})$ <hr/> Sufficient (cf. Theorem 3.10) G generic \mathbf{A} s with $m > k + n/G$	[87]
(ESPLIT)	Necessary Single forward operator \mathbf{A} $p(\mathbf{x})$ invariant to transforms $\{\mathbf{T}_g\}_{g=1}^G$ $\sum_{g=1}^G \mathbf{T}_g \mathbf{A}^\top \mathbf{A} \mathbf{T}_g$ is invertible <hr/> Sufficient $\mathbb{E}_{\mathbf{A}_1 \mathbf{A}} \left\{ \sum_{g \in \mathcal{S}_{\mathbf{A}_1}} \mathbf{T}_g^\top \mathbf{A}^\top \mathbf{A} \mathbf{T}_g \right\}$ is invertible where $\mathcal{S}_{\mathbf{A}_1} = \{g : \mathbf{A}_1 \text{ is a split of } \mathbf{AT}_g\}$	[90]
(EI)	Necessary Single forward operator \mathbf{A} $\text{supp}(p_{\mathbf{x}})$ invariant to transforms $\{\mathbf{T}_g\}_{g=1}^G$ $\sum_{g=1}^G \mathbf{T}_g \mathbf{A}^\top \mathbf{A} \mathbf{T}_g$ is invertible $\exists f^*$ such that $\mathbf{x} \approx f^*(\mathbf{Ax})$ <hr/> Sufficient (cf. Theorem 3.11) Generic \mathbf{A} with $m > 2k + n/G + 1$	[26, 62] [91, 101]

Table 3.1: **Summary of losses for learning from incomplete measurements.** The first two losses rely on having measurements with multiple operators, whereas the last two assume that the signal distribution is invariant to a group of transformations to obtain a set of virtual operators.

Chapter 4

Finite dataset effects

So far we have seen how to build self-supervised losses that are unbiased estimators of the (constrained or unconstrained) supervised loss in expectation. However, an important question remains, how good are the approximations with a finite number of samples? In this chapter, we discuss existing answers, while noting that a full theoretical characterization of the sample complexity of self-supervised methods is not yet fully understood. We illustrate the dependency of self-supervised methods on the dataset size in some practical scenarios, showing that it typically scales similarly to the supervised setting. We also introduce some practical tools for dealing with finite datasets: applying the hold-out method to avoid under or overfitting, and starting from pretrained models to reduce the number of measurements required to obtain good performances.

4.1 Hold-out method with self-supervised losses

A standard practice in machine learning is to divide the dataset into non-overlapping training, validation and testing sets [102]. The validation set plays a crucial role to avoid under or overfitting: if the validation loss remains always close to the training one, the model might not be expressive enough to fit all the data, and on the contrary, if the validation loss is bigger than the training one, the model is probably overfitting the training set.

A similar practice can be done in the self-supervised setting, even if we do not have ground truth validation samples [75, 103]. Since self-supervised losses serve as a proxy for the supervised loss, they can also be used on a validation set without ground truth to verify if the model is under or overfitting the data.

Figure 4.1 shows a self-supervised loss on the training and validation sets, and the supervised loss on the test set. The self-supervised validation loss tracks very well the performance on the test set, and can be used to stop the training when the model starts overfitting, i.e., when the gap between validation and training increases.

4.2 Variance of the loss and its gradients

A first step towards understanding how well self-supervised losses approximate the supervised counterparts is to study the variance of the losses and the variance of their gradients. A larger variance means that we will have a worse estimation of the supervised loss, leading

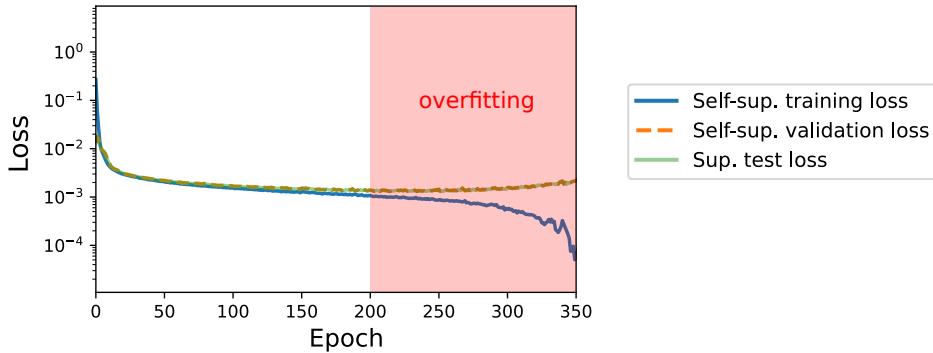


Figure 4.1: Training, validation and test losses using a UNet network on a Gaussian denoising problem with $\sigma = 0.1$. Training and validation are computed using the (SURE) loss presented in Chapter 2. Measurements are generated from the MNIST dataset, with 768 noisy images on the training set, 256 noisy images on the validation set, and 8192 (supervised pairs) of noisy and clean test images (not available in real-world settings). The self-supervised validation loss serves as a very accurate proxy for the supervised test error, and can be used to stop the training if the model starts overfitting the training data.

to a decrease in performance in comparison with the supervised case. Figure 4.2 shows the average normalized mean squared error for loss and gradient estimates for (Noise2Noise) and (SURE) using a DRUNet denoiser [104] on 512×512 patches corrupted by isotropic Gaussian noise using the Urban100 dataset. The experiment is repeated for a network with trained weights, and one with randomly initialized weights. In both trained and untrained cases, the error with respect to the supervised loss is below 30% for all noise levels. The gradient estimates are more accurate in the untrained model than the trained one. In the untrained case, all losses give errors of around 10%, as the loss is large, and the excess variance of self-supervised losses is negligible. In the trained case, the loss is small, and the additional variance of self-supervised losses starts to play a role. Noise2Noise gives better estimates than SURE, as it relies on more information, i.e., two independent noisy copies of each image.

When averaging over N noisy images, we should expect that the variances of all losses to decay as $1/N$ if the image samples are independent. This can give us an idea of *effective* sample complexity of a self-supervised method, by understanding how much larger N needs to be to match the variance of the supervised dataset. In the case of $\sigma = .5$ in Figure 4.2, we need approximately $\sqrt{10} \approx 3$ times more noisy samples to obtain the variance with SURE compared to the supervised case.

Variance of the loss In the following analysis, we focus on the (Noise2Noise) loss for simplicity, but the intuition carries over to the other losses presented in the previous chapters. An in-depth analysis of the variance of the (SURE) loss can be found in [105]. Due to the independence of \mathbf{y}_1 and \mathbf{y}_2 conditional on \mathbf{x} , the variance of the Noise2Noise loss admits the following decomposition:

Proposition 4.1. *Let \mathbf{y}_1 and \mathbf{y}_2 be two random independent random variables conditional*

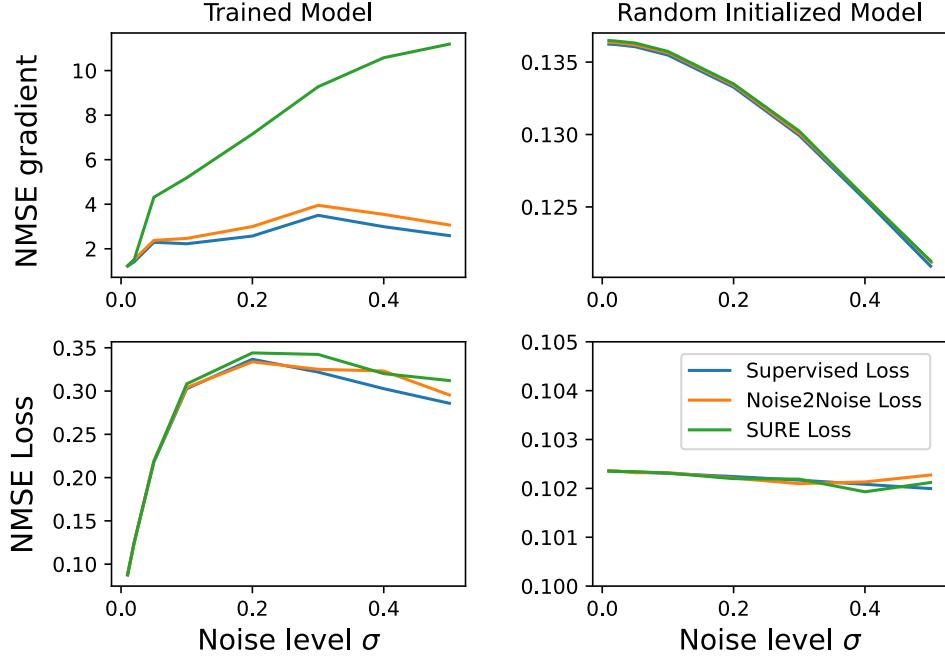


Figure 4.2: **Gradient approximation error of supervised and self-supervised losses.** Normalized mean squared error (NMSE) of the supervised, Noise2Noise and Monte Carlo SURE losses, and of its gradients with respect to the network weights. The experiments uses a DRUNet denoiser architecture evaluated on 512×512 patches of the Urban100 dataset.

on \mathbf{x} , such that $\mathbb{E}_{\mathbf{y}_2|\mathbf{x}} \{\mathbf{y}_2\} = \mathbf{x}$. Then

$$\mathbb{V}_{\mathbf{y}_1, \mathbf{y}_2} \{\mathcal{L}_{N2N}(\mathbf{y}_1, \mathbf{y}_2, f)\} = \mathbb{V}_{\mathbf{x}, \mathbf{y}_1} \left\{ \frac{1}{n} \|f(\mathbf{y}_1) - \mathbf{x}\|^2 \right\} + \Delta$$

where the first term is the variance of the supervised loss and Δ is the additional variance with respect to the supervised case, which is given by

$$\begin{aligned} \Delta &= \mathbb{V}_{\mathbf{x}, \mathbf{y}_2} \{\|\mathbf{y}_2 - \mathbf{x}\|^4\} + \frac{4}{n^2} \mathbb{E}_{\mathbf{x}} \{ \text{trace}(\boldsymbol{\Psi}_{\mathbf{x}} \boldsymbol{\Sigma}_{\mathbf{x}}) \} \\ &\quad - \frac{2}{n^2} \mathbb{E}_{\mathbf{x}} \left\{ \mathbb{E}_{\mathbf{y}_2|\mathbf{x}} \left\{ \|\mathbf{y}_2 - \mathbf{x}\|^2 (\mathbf{y}_2 - \mathbf{x})^\top \right\} \left(\mathbb{E}_{\mathbf{y}_1|\mathbf{x}} \{f(\mathbf{y}_1)\} - \mathbf{x} \right) \right\} \end{aligned}$$

where $\boldsymbol{\Sigma}_{\mathbf{x}} = \mathbb{E}_{\mathbf{y}_2|\mathbf{x}} \{(\mathbf{y}_2 - \mathbf{x})(\mathbf{y}_2 - \mathbf{x})^\top\}$ is the covariance of the noisy target \mathbf{y}_2 , and $\boldsymbol{\Psi}_{\mathbf{x}} = \mathbb{E}_{\mathbf{y}_1|\mathbf{x}} \{(f(\mathbf{y}_1) - \mathbf{x})(f(\mathbf{y}_1) - \mathbf{x})^\top\}$ is the error covariance for an image \mathbf{x} .

The proof is included in Appendix C. The last term in Δ is zero if the noise distribution is symmetric $\mathbb{E}_{\mathbf{y}_2|\mathbf{x}} \{\|\mathbf{y}_2 - \mathbf{x}\|^2 (\mathbf{y}_2 - \mathbf{x})\} = \mathbf{0}$, or if the estimator f is unbiased $\mathbb{E}_{\mathbf{y}_1|\mathbf{x}} \{f(\mathbf{y}_1)\} = \mathbf{x}$ for all \mathbf{x} . For example, in the simple case of targets with isotropic Gaussian noise, $\mathbf{y}_2 = \mathbf{x} + \sigma \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ we have

$$\Delta = \frac{3}{n} \sigma^4 + \frac{4\sigma^2}{n} \mathbb{E}_{\mathbf{x}, \mathbf{y}_1} \left\{ \frac{1}{n} \|f(\mathbf{y}_1) - \mathbf{x}\|^2 \right\}$$

which goes to zero as n grows, as long as the mean squared error, $\mathbb{E}_{\mathbf{x}, \mathbf{y}_1} \left\{ \frac{1}{n} \|f(\mathbf{y}_1) - \mathbf{x}\|^2 \right\}$, is approximately independent of n .

Variance of the gradients In order to compute the variance of the gradients, we need to consider a parameterization of the denoiser, f_{θ} , where $\theta \in \mathbb{R}^p$ are the trainable parameters of the denoiser (e.g., the network weights). The gradients could vary significantly even when the self-supervised loss has very small variance, if the denoiser is highly sensitive to changes in the parameters. The gradients of the (Noise2Noise) loss can be decomposed into two independent quantities as

$$\frac{\partial \mathcal{L}_{\text{N2N}}}{\partial \theta}(\mathbf{y}_1, \mathbf{y}_2, f_{\theta}) \propto \frac{1}{n} \frac{\partial f_{\theta}}{\partial \theta} (\mathbf{f}_{\theta}(\mathbf{y}_1) - \mathbf{y}_2) \quad (4.1)$$

$$\propto \underbrace{\frac{1}{n} \frac{\partial f_{\theta}}{\partial \theta} (\mathbf{f}_{\theta}(\mathbf{y}_1) - \mathbf{x})}_{\text{Supervised gradient}} - \underbrace{\frac{1}{n} \frac{\partial f_{\theta}}{\partial \theta} (\mathbf{y}_2 - \mathbf{x})}_{\text{Additional noise}} \quad (4.2)$$

where $\frac{\partial f_{\theta}}{\partial \theta} \in \mathbb{R}^{p \times n}$ is the Jacobian of the denoiser evaluated at \mathbf{y}_1 . The first term corresponds to the gradient of the supervised loss and the second term comprises the additional randomness due to the use of a noisy target. Since \mathbf{y}_1 and \mathbf{y}_2 are independent conditional on \mathbf{x} , the variance of the loss gradient is given by

$$\begin{aligned} & \mathbb{V}_{\mathbf{y}_1, \mathbf{y}_2} \left\{ \left\| \frac{\partial \mathcal{L}_{\text{N2N}}}{\partial \theta}(\mathbf{y}_1, \mathbf{y}_2, f_{\theta}) \right\|^2 \right\} \\ &= \mathbb{V}_{\mathbf{y}_1, \mathbf{x}} \left\{ \left\| \frac{1}{n} \frac{\partial f_{\theta}}{\partial \theta} (\mathbf{f}_{\theta}(\mathbf{y}_1) - \mathbf{x}) \right\|^2 \right\} + \mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}} \left\{ \left\| \frac{1}{n} \frac{\partial f_{\theta}}{\partial \theta} (\mathbf{y}_2 - \mathbf{x}) \right\|^2 \right\} \\ &= \underbrace{\mathbb{V}_{\mathbf{y}_1, \mathbf{x}} \left\{ \left\| \frac{1}{n} \frac{\partial f_{\theta}}{\partial \theta} (\mathbf{f}_{\theta}(\mathbf{y}_1) - \mathbf{x}) \right\|^2 \right\}}_{\text{Variance of supervised loss}} + \underbrace{\frac{1}{n^2} \mathbb{E}_{\mathbf{x}} \left\{ \text{trace} \left(\Sigma_{\mathbf{x}} \mathbb{E}_{\mathbf{y}_1 | \mathbf{x}} \left\{ \frac{\partial f_{\theta}}{\partial \theta} \frac{\partial f_{\theta}}{\partial \theta}^\top \right\} \right) \right\}}_{\text{Additional variance}} \end{aligned}$$

where the second line uses the decomposition in (4.2) and that the additional noise has zero mean, and the third line defines $\Sigma_{\mathbf{x}} := \mathbb{E}_{\mathbf{y}_2 | \mathbf{x}} \{(\mathbf{y}_2 - \mathbf{x})(\mathbf{y}_2 - \mathbf{x})^\top\}$ as the covariance of the noisy target \mathbf{y}_2 . For example, in the case of targets with isotropic Gaussian noise, $\mathbf{y}_2 = \mathbf{x} + \sigma \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ we have

$$\text{Additional variance} = \sigma^2 \mathbb{E}_{\mathbf{y}_1} \left\{ \left\| \frac{1}{n} \frac{\partial f_{\theta}}{\partial \theta} \right\|_F^2 \right\}$$

where the second term is the Frobenius norm of the Jacobian of the network with respect to its parameters. The additional variance introduced by the noisy target \mathbf{y}_2 depends on the noise level σ^2 , and the average sensitivity of the network's output to changes in the input weights.

4.3 Gap with supervised learning

A key question when comparing supervised and self-supervised methods, is how self-supervised methods compare with supervised counterparts as a function of the amount of data we have for training, which amounts to computing the following gap:

$$\text{gap}(N) = \min_{\theta} \frac{1}{N} \sum_{i=1}^N \frac{1}{n} \|f_{\theta}(\mathbf{y}_{1,i}) - \mathbf{y}_{2,i}\|^2 - \underbrace{\min_{\theta} \mathbb{E}_{\mathbf{x}, \mathbf{y}_1} \left\{ \frac{1}{n} \|f_{\theta}(\mathbf{y}_1) - \mathbf{x}\|^2 \right\}}_{\approx \text{MMSE}}$$

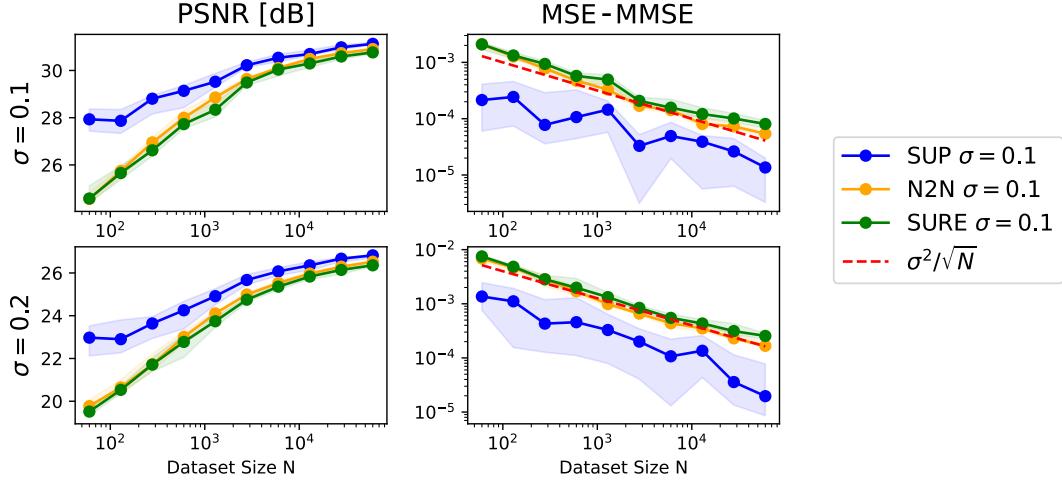


Figure 4.3: **Gap of Noise2Noise and SURE w.r.t. supervised learning as a function of dataset size for MNIST Gaussian denoising using a U-Net denoiser.** On the left, we show the PSNR obtained by each learning method, and on the right we show the test mean squared error gap compared to the supervised baseline using the full dataset. The experiment is repeated 15 times for each dataset size N and each of the two different noise levels $\sigma_1 = .1$ and $\sigma_2 = .2$, with shaded areas denoting the 90% intervals across repetitions. The optimality gap follows approximately σ^2/\sqrt{N} .

Quantifying this gap is generally a difficult problem, since the results can be highly dependent on the data distribution, the parameterization of the estimator, and the specific learning algorithm used for estimating the parameters. Understanding the sample complexity of learning methods is an active area of research [102] with many open questions [106]. In particular, a challenging problem is to obtain meaningful bounds that are not highly dependent on the number of parameters of the model, which is typically very large in deep networks. For example, a line of work [103, 107] studies the generalization error of stochastic gradient descent methods, which are the most popular optimization methods for training deep networks, obtaining bounds that are approximately independent of the parameter count.

Here, we provide an empirical evaluation of the sample complexity, using the (self-supervised) hold-out method described in Section 4.1 to obtain the best model for each dataset size. Figure 4.3 shows the empirical gap for the networks trained with Noise2Noise or SURE on an MNIST Gaussian denoising problem for different dataset sizes. The observed gap follows the asymptotic behaviour $\text{gap}(N) \propto \sigma^2/\sqrt{N}$. How good is this rate? We can gain some intuition by comparing it with the setting of a trivial signal distribution consisting of a single signal: in this case, we could estimate the signal by simply averaging N noisy realizations, and we would get the standard rate for estimating the mean of a Gaussian distribution of variance σ^2 , that is σ^2/N . Thus, the cost of dealing with non-trivial signal distributions is a factor of \sqrt{N} .

4.4 Fine-tuning and test-time adaptation

While most self-supervised losses presented in this manuscript approximate the supervised loss and can be used to train a reconstruction network from a random initialization without any ground truth references, they can also be used for fine-tuning a pretrained network on new measurement data, which might differ from the supervised dataset used for pretraining. This procedure is often referred to as test-time adaptation [108].

Starting from a pretrained model can significantly reduce the number of samples and training time needed to obtain good results compared with a randomly initialized model (e.g., often a couple of samples suffices). Moreover, fine-tuning can significantly improve the performance of the pretrained model on out-of-domain measurement data [17].

Chapter 5

Extensions and open problems

The ideas of self-supervised learning are already making a considerable impact in imaging and sensing, with applications emerging in MRI [77], microscopy [48], and remote sensing [49] to name but a few. In this final chapter, we review some ongoing work that is exploring extensions of the self-supervised learning framework for inverse problems and present some of the open problems in the field.

5.1 Non-linear inverse problems

Many real-world inverse problems are non-linear, such as quantized sensing [109], phase retrieval [110], and non-linear problems associated to partial differential equations, such as the inverse scattering problem [8]. While, in principle, most of the self-supervised losses presented in Chapter 3 can be applied with non-linear forward models, most of the theoretical analyses associated with these losses are restricted to the linear case and the development of a general theoretical framework for nonlinear operators is an open problem.

Another important challenge in non-linear settings is that self-supervised losses involve the evaluation of the non-linear operator \mathbf{A} . This can both be computationally expensive, and lead to more complex loss landscapes including many local minima, compared to the supervised loss that does not involve the operator \mathbf{A} . We believe that these problems introduce new challenges, such as exploring relaxations of \mathbf{A} in the self-supervised loss [111, 112] and problem-specific optimization algorithms.

Hu et al. [113] show that the (multi-operator) splitting loss can be used in the context of accelerated MRI with unknown coil maps, which can be seen as a bilinear inverse problem.

The equivariant imaging approach has also been extended to declipping problems given by $\mathbf{A}(\mathbf{x}) = \eta(\mathbf{x})$ where $\eta : \mathbb{R} \rightarrow [-1, 1]$ is an elementwise clipping operator. In this case, learning beyond the clipping threshold is possible if we can assume that the signal model is invariant to amplitude scaling, $\mathbf{T}_g = g\mathbf{I}$ with $g > 0$, as the forward operator is not equivariant to these transformations [112].

The model identification theory in Section 3.4 has been extended to quantized inverse problems [111], in the extreme case where every measurement is quantized to a single bit, a problem that can be written as $\mathbf{A}_g(\mathbf{x}) = \text{sign}(\mathbf{Q}\mathbf{x})$ where $\{\mathbf{Q}_g \in \mathbb{R}^{m \times n}\}_{g=1}^G$ are linear operators. In this case, exact identification of the support of the signal distribution is impossible even if the set has low dimension $k \ll n$, however it is still possible to learn an approximation

up to a global error of order $\mathcal{O}(\frac{k+n/G}{m} \log \frac{nm}{k+n/G})$.

5.2 Towards large scale self-supervised imaging

Deep learning imaging solutions rely on the substantial computing power offered by modern GPUs. Although this technology is advancing at a rapid pace, the current memory capacities of consumer GPUs limit the size of imaging inverse problems that can be handled. For example, this is the case for challenging high-dimensional medical imaging such as, extreme scale 3D or 4D (3D + time) CT and MRI imaging [114–116], as well as applications like ptychography in electron microscopy [117]. Such problems can generate as much as 10s of gigabytes of measurements and/or image data *per reconstruction*, with the size of both image and measurements easily surpassing the memory capabilities of most GPUs.

Training deep learning solutions for such problems therefore faces additional computational complications. This is made all the more challenging when looking to use self-supervised learning techniques as the associated loss functions require the calculation of (and backpropagation through) the forward operator, \mathbf{A} , resulting in both large computation and memory usage in training and possible also in evaluation. The same issue occurs when training unrolled deep learning solutions in the supervised scenario, e.g., [114, 116], where various approaches have been considered to mitigate these computational issues, e.g., forward or reverse recalculation, and gradient checkpointing for reduced storage during backpropagation, and data splitting methods which have traditionally been used in model-based image reconstruction [118, 119].

Understanding the best approaches for tackling these issues in self-supervised learning has received much less attention and is a fruitful area for future research. One exception is [120], where the authors train an image reconstruction network for low-dose 3D helical CT in a self-supervised manner. Their core approach was to use a measurement splitting technique similar in spirit to the Noise2Inverse method [25] that only involved partial calculation of the forward operator at each iteration (they also implemented a range of other computational tricks such as gradient checkpointing, customised CUDA modules, etc.).

While Kosoma et al. [120] focused on an invertible imaging operator, it should be straightforward to extend these ideas to non-invertible multiple operators using the related splitting ideas presented in Chapter 3. Another interesting direction is exploring self-supervised learning solutions that explicitly use reconstruction networks based on stochastic optimisation [121].

5.3 Robust solutions and partially defined models

Another practical issue that is important to address in real world inverse problems is the accuracy with which we can define the observation model. As George Box said “all models are wrong, but some are useful.” In any statistical learning or inference scenario it is important to capture as accurately as possible the underlying relationships between the variables, but also to ensure that the solutions take account of any unknown components of the forward model and are robust to any approximations/imperfections. This is particularly important in self-supervised learning, where the surrogates for the supervised loss rely heavily on the additional information from the inverse problem.

In Chapter 2, we have already seen instances of this that took account of partially defined noise models - constraining the class of estimators to be learned, e.g., [21, 28, 75]. This typically leads to sub-optimal solutions that nevertheless can outperform "optimal" solutions with misspecified assumptions.

Extending these ideas to partially defined or misspecified forward models, either theoretically and algorithmically, is much more challenging and is an interesting direction for future research. For example, partially defined models include the case when there are unknown calibration parameters that must be estimated. One approach to this is to treat the unknown parameters as additional unknowns within the imaging problem that can either be estimated or marginalised out as part of the reconstruction process, e.g. self-supervised estimation of coil sensitivity maps in MRI [113], or self-supervised blind deblurring solutions [122].

A common source of approximation within the forward model is the typical digital representation of the continuous image as a finite number of pixels/voxels. Acquisition systems are often designed to avoid introducing aliasing within such digital representations which in turn can induce correlation within the image noise which in the self-supervised learning setting must be treated with care, e.g., in SAR processing [49].

5.4 Beyond the ℓ_2 loss

Most of the losses presented in this manuscript aim at approximating the supervised ℓ_2 loss. This loss has the benefit of having a simple decomposition

$$\|f(\mathbf{y}) - \mathbf{x}\|^2 = \|f(\mathbf{y}) - \mathbf{y}\|^2 - 2 f(\mathbf{y})^\top (\mathbf{y} - \mathbf{x}) + \text{const.}$$

where the second term captures the difference between the supervised case with simple measurement consistency. As we have seen in Chapter 2, this term can be handled using independent noise realizations as in (Noise2Noise) and (R2R), Stein's lemma as in (SURE), or blind-spot networks as in (CV). While similar expressions also hold for Bregman divergences [47, 71], this decomposition does not apply on other popular losses, such as the ℓ_1 or ℓ_0 loss.

In Noise2Noise [40], general ℓ_p losses are proposed for handling noise distributions with non zero-mean noise, such as the ℓ_1 loss for random text removal or the ℓ_0 loss for salt-and-pepper noise. Recalling that Noise2Noise relies on independent noisy pairs $(\mathbf{y}_1, \mathbf{y}_2)$, training on an ℓ_0 loss leads (in expectation) to the mode estimator

$$\text{Mode}\{\mathbf{y}_2|\mathbf{y}_1\} = \arg \max_{\mathbf{y}_2} p(\mathbf{y}_2|\mathbf{y}_1).$$

In general, $\text{Mode}\{\mathbf{y}_2|\mathbf{y}_1\} \neq \text{Mode}\{\mathbf{y}_2|\mathbf{x}\}$, even if \mathbf{y}_2 and \mathbf{y}_1 are independent given \mathbf{x} . However, under the assumption that the (posterior) distribution of \mathbf{x} given \mathbf{y}_1 is heavily concentrated, we have $p(\mathbf{y}_2|\mathbf{y}_1) \approx p(\mathbf{y}_2|\mathbf{x})$, and thus $\text{Mode}\{\mathbf{y}_2|\mathbf{y}_1\} \approx \text{Mode}\{\mathbf{y}_2|\mathbf{x}\} = \mathbf{x}$. A similar argument holds for the ℓ_1 loss and noise distributions whose median is given by \mathbf{x} . While this approximation provides good empirical results [40], a better understanding under which conditions such approximations are reasonable, and determining how to extend (SURE) and other noise distribution aware self-supervised losses to general ℓ_p losses, are interesting directions of future research.

5.5 Uncertainty quantification and generative modelling

Most of the focus in this review has been on learning a good reconstruction mapping or denoiser, often targeting the conditional mean of the inverse problem, $\mathbb{E}_{\mathbf{x}|\mathbf{y}} \{\mathbf{x}\}$. However, in many scenarios it is also important to quantify the uncertainty for a given estimator, so that the image estimate can be used with confidence in downstream analysis.

Most self-supervised losses presented in this manuscript serve as estimators of the supervised loss, and can thus be used to estimate reconstruction errors at test time. For example, (SURE) or (R2R) can be used to estimate the mean squared error, and higher-order extensions such as SURE for SURE [105] can be used to quantify the uncertainty of this error estimate. It is also possible to use extensions of (Tweedie) to high order moments of the posterior distribution. For example, in the Gaussian denoising scenario, Manor and Michaeli [123] use these extensions to estimate the principal components for the covariance of the posterior distribution, $p(\mathbf{x}|\mathbf{y})$, directly from the MMSE estimator (which itself can be estimated in a self-supervised manner). This nicely augments the estimated denoised images with the principal directions of uncertainty. When dealing with incomplete data, (EI) can be used to quantify the reconstruction error in the nullspace of the forward operator [124].

The extent to which these ideas could be applied to other self-supervised learning solutions in this review is an interesting open problem.

Generative models While predicted error covariance of point estimates provide an efficient and compact characterization of the uncertainty, in certain areas of imaging science it is desirable to be able to explore the full posterior distribution of the imaging problem for downstream analysis, e.g., to characterize plausible solutions when ambiguities exist, or to test statistical hypotheses.

A popular machine learning solution in such circumstances is to learn a generative model, such as VAEs [125], GANs [126] or diffusion models [41] that can act as a stochastic simulator and provide samples from the posterior distribution, $p(\mathbf{x}|\mathbf{y})$. As such, generative models have become a popular approach for solving imaging inverse problems. However, in general such solutions currently rely on a *pre-trained* generative models that have been trained on existing ground truth data. An interesting research direction is therefore to what extent generative models can be learned in a purely self-supervised manner. Some progress has already been made on this. For example, as discussed in Chapters 2 and 3, Prakash et al. [80, 81] have developed VAEs for the Gaussian denoising problem, while GANs [82] and diffusion models [43, 85] have been proposed that can be trained from noiseless but incomplete measurements. However, these methods generally rely on low-noise measurements from multiple operators, and it remains an open question as to whether generative models could be trained with noisy measurements taken from a single ill-posed measurement operator in a similar manner to (EI).

5.6 Sample complexity

Most of the theoretical analysis of self-supervised learning imaging solutions are either geometric [111] or focus on the asymptotic properties of the learning problem considering the behaviour of expected values. However, these do not indicate how hard the problem is sta-

tistically in terms of the number of measurement training samples required to achieve a good solution. This essentially comes down to how accurately we can approximate the expected risk from the empirical risk.

We discussed this briefly in Chapter 4 with respect to Noise2Noise and showed empirically that the gap between supervised learning and the equivalent (in expectation) self-supervised learning strategy scales approximately as $\text{gap}(N) \propto \sigma^2/\sqrt{N}$. However, analyzing sample complexity, even in the supervised case, is a challenging problem [102]. Most existing results typically make significant simplifying assumptions, such as that the learning problem is convex [102], or that the reconstruction estimator is linear [103]. There is also the question of how the sample complexity behaves as a function of the image size. Here, we have reason to be optimistic that we may be able to benefit from a blessing of dimensionality [127] associated with the typical high dimensionality of images, in the similar manner to how we can get accurate estimates of the SURE loss using only a single Monte Carlo sample [70].

Understanding the nature of this supervised-self-supervised gap would help imaging practitioners to understand whether it is better to try to collect a small amount of supervised training data or whether the same result can be achieved through using a larger collection of measurement data that is usually much easier to acquire.

5.7 Choosing the right self-supervised method

In this article we have covered various self-supervised techniques offering capabilities ranging from denoising to solving ill-posed imaging inverse problems. We have also seen that the same problem can sometimes be solved through judicious choice of network architecture (but avoiding appealing to more nebulous architectural inductive biases) or through a cleverly designed loss function. However, we have refrained from explicitly promoting one solution over another. This may well leave the practitioner somewhat frustrated with a lack of guidance as to what is the right solution for a given task.

Depending on the scenario various forms of information may be available to the practitioner, e.g., in terms of the nature of and information about the measurement noise, or the number and type of measurement operators available for creating training data. In some instances this naturally selects subsets of relevant algorithms and techniques and these have been highlighted in the tables at the end of Chapters 2 and 3. However, it is important to also note that not all relevant algorithms make the same use of the available information. For example, if presented with a problem where pairs of noisy realizations of the same signal are available for training and testing, one might be tempted to naturally reach for the Noise2Noise algorithms. However, if one knows something about the statistical noise model, even if this is only partial, it may be better to simply average the multiple realizations (thereby gaining 3 dB of SNR) and applying a different technique that incorporates additional statistical information¹.

The fact that many of the reviewed techniques exploit different information also opens up the opportunity to explore hybrid approaches. For instance, in the example above, rather than simply averaging the noise image pairs and then applying a single image technique that incorporated further knowledge of the noise model, one could explicitly construct a version

¹For example, Tachella et al. [28] show that one can obtain better performance using UNSURE than Noise2Noise in a cryogenic electron microscopy denoising, where the noise model is approximately known.

that can exploit pairs of images². Similarly, one is not restricted to incorporating invariance properties into the inverse problem only when there is a single measurement operator. Indeed, empirical evidence suggests that exploiting multiple sources of information within the measurements tends to only make things better [128].

In many real world settings, the most challenging aspect for the practitioner is to know the accuracy of the underlying forward model that is being exploited to enable self-supervised learning, as we have discussed above. This is extremely important and should probably be the first thing for the practitioner to consider when selecting the right algorithm as better defined forward models typically offer better performance but at a price of being more sensitive to any misspecifications.

Acknowledgements

The authors are grateful for discussions with Dongdong Chen, Laurent Jacques, Marcelo Pereyra, Loïc Denis, Andrew Wang, Victor Sechaud, Jérémie Scanvic, César Caiafa, Thomas Davies, and Brett Levac. J. Tachella acknowledges support by the French National Research Agency (ANR-23-CE23-0013). The authors thank ENS Lyon for supporting the research visit of Mike Davies as invited professor.

²Note having access to pairs of noisy images immediately provides an estimate for the SNR of the signal.

Appendix A

Noisier2Noise and R2R equivalence

In this appendix, we show the asymptotic equivalence between the Noisier2Noise [51] and Recorrupted2Recorrupted [52] losses for the Gaussian denoising case, that is $\mathbf{y} = \mathbf{x} + \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \Sigma)$. Noisier2Noise proposes to train a network

$$\mathcal{L}_{\text{Noisier2Noise}}(\mathbf{y}, f) = \mathbb{E}_{\mathbf{y}_1|\mathbf{y}} \{ \|f(\mathbf{y}_1) - \mathbf{y}\|^2 \} \quad (\text{A.1})$$

where $\mathbf{y}_1 = \mathbf{y} + \tau\omega$ with $\omega \sim \mathcal{N}(\mathbf{0}, \Sigma)$ and $\tau > 0$. The minimizer of this loss in expectation is

$$\begin{aligned} f^*(\mathbf{y}_1) &= \mathbb{E}_{\mathbf{y}|\mathbf{y}_1} \{ \mathbf{y} \} \\ &= \mathbb{E}_{\mathbf{x}, \epsilon|\mathbf{y}_1} \left\{ \frac{\tau}{1+\tau} \mathbf{x} + \frac{1}{1+\tau} \mathbf{x} + \epsilon \right\} \\ &= \frac{\tau}{1+\tau} \mathbb{E}_{\mathbf{x}|\mathbf{y}_1} \{ \mathbf{x} \} + \frac{1}{1+\tau} \left(\mathbb{E}_{\mathbf{x}|\mathbf{y}_1} \{ \mathbf{x} \} + \mathbb{E}_{\epsilon|\mathbf{y}_1} \{ \epsilon \} + \frac{1}{1+\tau} \mathbb{E}_{\epsilon|\mathbf{y}_1} \{ \epsilon \} \right) \\ &= \frac{\tau}{1+\tau} \mathbb{E}_{\mathbf{x}|\mathbf{y}_1} \{ \mathbf{x} \} + \frac{1}{1+\tau} \left(\mathbb{E}_{\mathbf{x}|\mathbf{y}_1} \{ \mathbf{x} \} + \mathbb{E}_{\epsilon|\mathbf{y}_1} \{ \epsilon \} + \tau \mathbb{E}_{\omega|\mathbf{y}_1} \{ \omega \} \right) \\ &= \frac{\tau}{1+\tau} \mathbb{E}_{\mathbf{x}|\mathbf{y}_1} \{ \mathbf{x} \} + \frac{1}{1+\tau} \mathbb{E}_{\mathbf{x}|\mathbf{y}_1} \{ \mathbf{y}_1 \} \\ &= \frac{\tau}{1+\tau} \mathbb{E}_{\mathbf{x}|\mathbf{y}_1} \{ \mathbf{x} \} + \frac{1}{1+\tau} \mathbf{y}_1 \end{aligned}$$

where the third line uses that $\mathbb{E}_{\epsilon|\mathbf{y}_1} \{ \mathbf{x} \} = \mathbb{E}_{\omega|\mathbf{y}_1} \{ \omega \}$ since ϵ and ω are iid. This requires knowing the distribution of the noise for this result to hold.

$$f^{\text{test}}(\mathbf{y}_1) = \frac{1+\tau}{\tau} f^*(\mathbf{y}_1) - \frac{1}{\tau} \mathbf{y}_1 \quad (\text{A.2})$$

$$= \mathbb{E}_{\mathbf{x}|\mathbf{y}_1} \{ \mathbf{x} \} \quad (\text{A.3})$$

The (R2R) loss is defined as

$$\mathcal{L}_{\text{R2R}}(\mathbf{y}, f) = \mathbb{E}_{\mathbf{y}_1|\mathbf{y}} \{ \|f(\mathbf{y}_1) - \mathbf{y}_2\|^2 \} \quad (\text{A.4})$$

where $\mathbf{y}_1 = \mathbf{y} + \tau\omega$ and $\mathbf{y}_2 = \mathbf{y} - \frac{1}{\tau}\omega$, with $\omega \sim \mathcal{N}(\mathbf{0}, \Sigma)$ and $\tau > 0$. As we show in Chapter 2, this loss is an unbiased estimator of the supervised ℓ_2 loss with input \mathbf{y}_1 , and thus its minimizer is $f^*(\mathbf{y}_1) = \mathbb{E}_{\mathbf{x}|\mathbf{y}_1} \{ \mathbf{x} \}$ which is the same as the Noisier2Noise test time function in (A.3).

Appendix B

Identification of moments

Recovering the signal distribution, $p_{\mathbf{x}}$, from the measurement distribution, $p_{\mathbf{y}}$, can be seen as an inverse problem in infinite dimensions defined by the forward problem

$$p_{\mathbf{y}}(\mathbf{y}) = \int_{\mathbf{x} \in \mathcal{X}} p(\mathbf{y}|\mathbf{x})p_{\mathbf{x}}(\mathbf{x})d\mathbf{x}. \quad (\text{B.1})$$

Identifying $p_{\mathbf{x}}$ from $p_{\mathbf{y}}$ is possible if the noise distribution has a nowhere zero characteristic function (see Section 2.4), and, in the case of incomplete observations from multiple operators, if the support of $p_{\mathbf{x}}$ is low-dimensional (see Section 3.4).

However, in some cases, we might not be able to identify the full distribution $p_{\mathbf{x}}$, but we can still find some of its higher-order moments, or moments of the posterior distribution $p(\mathbf{x}|\mathbf{y})$.

Example B.1. Assume a Bernoulli noise model $\mathbf{y}|\mathbf{x} \sim \text{Ber}(\mathbf{x})$ where $p(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^n x_i^{y_i} (1-x_i)^{1-y_i}$. Since measurements are binary, we identify at most $2^n - 1$ different moments of $p_{\mathbf{x}}$ associated with all possible inputs of $p_{\mathbf{y}}$, which are given by $\mathbb{E}_{\mathbf{x}} \{\prod_{i \in \mathcal{I}} x_i\}$ where \mathcal{I} is an arbitrary choice of indices in $\{1, \dots, n\}$.

Since we can compute any moment of $p_{\mathbf{y}}$, using the law of total expectation we have that

$$\mathbb{E}_{\mathbf{y}} \{g(\mathbf{y})\} = \mathbb{E}_{\mathbf{x}} \{\mathbb{E}_{\mathbf{y}|\mathbf{x}} \{g(\mathbf{y})\}\} \quad (\text{B.2})$$

$$= \mathbb{E}_{\mathbf{x}} \{r(\mathbf{x})\} \quad (\text{B.3})$$

where we defined $r(\mathbf{x}) := \mathbb{E}_{\mathbf{y}|\mathbf{x}} \{g(\mathbf{y})\}$ for some function $g : \mathbb{R}^m \mapsto \mathbb{R}$.

In the case where $p_{\mathbf{y}}$ is continuous and differentiable, we can compute moments of $p_{\mathbf{x}}$ by differentiating (B.1) as

$$\frac{\partial^k p(\mathbf{y})}{\partial y_i^k} = \mathbb{E}_{\mathbf{x}} \left\{ \frac{\partial^k p(\mathbf{y}|\mathbf{x})}{\partial y_i^k} \right\} \quad (\text{B.4})$$

$$= \mathbb{E}_{\mathbf{x}} \{\tilde{r}_{i,k}(\mathbf{x}, \mathbf{y})\} \quad (\text{B.5})$$

for any $k \geq 0$ and $i = 1, \dots, n$, where we defined $\tilde{r}_{i,k}(\mathbf{x}, \mathbf{y}) := \frac{\partial^k p(\mathbf{y}|\mathbf{x})}{\partial y_i^k}$.

Example B.2. Consider a Gaussian noise model where we have $p(\mathbf{y}|\mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$, such that $\tilde{r}_{i,1}(\mathbf{x}, \mathbf{y}) = \frac{y_i - x_i}{\sigma^2} p(\mathbf{y}|\mathbf{x})$. Using this result, we obtain

$$\mathbb{E}_{\mathbf{x}} \left\{ \frac{y_i - x_i}{\sigma^2} p(\mathbf{y}|\mathbf{x}) \right\} = \frac{y_i}{\sigma^2} p(\mathbf{y}) - \frac{1}{\sigma^2} \mathbb{E}_{\mathbf{x}} \{x_i p(\mathbf{y}|\mathbf{x})\} \quad (\text{B.6})$$

$$\frac{\partial p(\mathbf{y})}{\partial y_i} = \frac{y_i}{\sigma^2} p(\mathbf{y}) - \frac{1}{\sigma^2} \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{x_i\} p(\mathbf{y}) \quad (\text{B.7})$$

$$\sigma^2 \frac{\partial \log p(\mathbf{y})}{\partial y_i} = y_i - \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{x_i\} \quad (\text{B.8})$$

which is the well-known (Tweedie) formula, $\mathbb{E}\{\mathbf{x}|\mathbf{y}\} = \mathbf{y} + \sigma^2 \nabla \log p_{\mathbf{y}}(\mathbf{y})$, the same formula that we derived in Section 2.2.2 as the minimizer (in expectation) of the (SURE) loss. Higher order derivatives can be used to estimate higher posterior moments [123], such as the posterior variance which is equivalent to the minimum mean square error:

$$MMSE = \sigma^2 \left(\frac{1}{n} \sum_{i=1}^n \frac{\partial^2 \log p_{\mathbf{y}}}{\partial y_i^2}(\mathbf{y}) \right).$$

Appendix C

Additional proofs

Proposition C.1. Let $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$ be two random variables following the joint distribution $p(\mathbf{x}, \mathbf{y})$. The minimizer of the following ℓ_2 loss

$$f^*(\mathbf{y}) = \arg \min_f \mathbb{E}_{\mathbf{x}, \mathbf{y}} \{ \|f(\mathbf{y}) - \mathbf{x}\|^2 \} \quad (\text{C.1})$$

is given by

$$f^*(\mathbf{y}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ \mathbf{x} \}. \quad (\text{C.2})$$

Proof. Letting $\mathcal{L}(f) = \mathbb{E}_{\mathbf{x}, \mathbf{y}} \{ \|f(\mathbf{y}) - \mathbf{x}\|^2 \}$, we have that

$$\begin{aligned} \mathcal{L}(f) &= \mathbb{E}_{\mathbf{x}, \mathbf{y}} \{ \| (f(\mathbf{y}) - f^*(\mathbf{y})) - (\mathbf{x} - f^*(\mathbf{y})) \|^2 \} \\ &\propto \mathbb{E}_{\mathbf{x}, \mathbf{y}} \{ \|f(\mathbf{y}) - f^*(\mathbf{y})\|^2 \} - 2 \mathbb{E}_{\mathbf{x}, \mathbf{y}} \left\{ (f(\mathbf{y}) - f^*(\mathbf{y}))^\top (\mathbf{x} - f^*(\mathbf{y})) \right\} \\ &= \mathbb{E}_{\mathbf{x}, \mathbf{y}} \{ \|f(\mathbf{y}) - f^*(\mathbf{y})\|^2 \} - 2 \mathbb{E}_{\mathbf{y}} \left\{ \mathbb{E}_{\mathbf{x}|\mathbf{y}} \left\{ (f(\mathbf{y}) - f^*(\mathbf{y}))^\top (\mathbf{x} - f^*(\mathbf{y})) \right\} \right\} \\ &= \mathbb{E}_{\mathbf{x}, \mathbf{y}} \{ \|f(\mathbf{y}) - f^*(\mathbf{y})\|^2 \} - 2 \mathbb{E}_{\mathbf{y}} \left\{ (f(\mathbf{y}) - f^*(\mathbf{y}))^\top \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ \mathbf{x} - f^*(\mathbf{y}) \} \right\} \\ &= \mathbb{E}_{\mathbf{x}, \mathbf{y}} \{ \|f(\mathbf{y}) - f^*(\mathbf{y})\|^2 \} \end{aligned}$$

where the fourth line uses the fact that $\mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ \mathbf{x} - f^*(\mathbf{y}) \} = 0$. Thus, the global minimizer of $\mathcal{L}(f)$ is $f(\mathbf{y}) = f^*(\mathbf{y}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ \mathbf{x} \}$. \square

Proposition C.2. Let $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ be two random variables following the joint distribution $p(\mathbf{x}, \mathbf{y})$, and let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a linear operator. The minimizer of the following weighted ℓ_2 loss

$$f^*(\mathbf{y}) \in \arg \min_f \mathbb{E}_{\mathbf{x}, \mathbf{y}} \{ \| \mathbf{A}f(\mathbf{y}) - \mathbf{Ax} \|^2 \} \quad (\text{C.3})$$

is given by

$$f^*(\mathbf{y}) = \mathbf{A}^\dagger \mathbf{A} \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ \mathbf{x} \} + (\mathbf{I} - \mathbf{A}^\dagger \mathbf{A})v(\mathbf{y}) \quad (\text{C.4})$$

where \mathbf{A}^\dagger is the linear pseudoinverse of \mathbf{A} , $\mathbf{A}^\dagger \mathbf{A}$ is the projection into the range space of \mathbf{A}^\top , and $v : \mathbb{R}^n \mapsto \mathbb{R}^n$ is any function.

Proof. Defining $\tilde{\mathbf{x}} = \mathbf{Ax}$ and $\tilde{f} = \mathbf{A} \circ f$, we can apply Proposition C.1 to conclude that $\tilde{f}^*(\mathbf{y}) = \mathbb{E}_{\tilde{\mathbf{x}}|\mathbf{y}} \{ \tilde{\mathbf{x}} \}$, or equivalently that $\mathbf{A}f^*(\mathbf{y}) = \mathbb{E}_{\mathbf{x}|\mathbf{y}} \{ \mathbf{Ax} \}$. Applying the linear pseudoinverse of \mathbf{A} on both sides, we obtain the desired equality in (C.4). \square

Proposition 4.1. Let \mathbf{y}_1 and \mathbf{y}_2 be two random independent random variables conditional on \mathbf{x} , such that $\mathbb{E}_{\mathbf{y}_2|\mathbf{x}}\{\mathbf{y}_2\} = \mathbf{x}$. Then

$$\mathbb{V}_{\mathbf{y}_1, \mathbf{y}_2}\{\mathcal{L}_{N2N}(\mathbf{y}_1, \mathbf{y}_2, f)\} = \mathbb{V}_{\mathbf{x}, \mathbf{y}_1}\left\{\frac{1}{n}\|f(\mathbf{y}_1) - \mathbf{x}\|^2\right\} + \Delta$$

where the first term is the variance of the supervised loss and Δ is the additional variance with respect to the supervised case, which is given by

$$\begin{aligned}\Delta &= \mathbb{V}_{\mathbf{x}, \mathbf{y}_2}\{\|\mathbf{y}_2 - \mathbf{x}\|^4\} + \frac{4}{n^2}\mathbb{E}_{\mathbf{x}}\{\text{trace}(\boldsymbol{\Psi}_{\mathbf{x}}\boldsymbol{\Sigma}_{\mathbf{x}})\} \\ &\quad - \frac{2}{n^2}\mathbb{E}_{\mathbf{x}}\left\{\mathbb{E}_{\mathbf{y}_2|\mathbf{x}}\left\{\|\mathbf{y}_2 - \mathbf{x}\|^2(\mathbf{y}_2 - \mathbf{x})^\top\right\}\left(\mathbb{E}_{\mathbf{y}_1|\mathbf{x}}\{f(\mathbf{y}_1)\} - \mathbf{x}\right)\right\}\end{aligned}$$

where $\boldsymbol{\Sigma}_{\mathbf{x}} = \mathbb{E}_{\mathbf{y}_2|\mathbf{x}}\{(\mathbf{y}_2 - \mathbf{x})(\mathbf{y}_2 - \mathbf{x})^\top\}$ is the covariance of the noisy target \mathbf{y}_2 , and $\boldsymbol{\Psi}_{\mathbf{x}} = \mathbb{E}_{\mathbf{y}_1|\mathbf{x}}\{(f(\mathbf{y}_1) - \mathbf{x})(f(\mathbf{y}_1) - \mathbf{x})^\top\}$ is the error covariance for an image \mathbf{x} .

Proof. Defining $a_{\mathbf{x}, \mathbf{y}_1} = \frac{1}{n}\|f(\mathbf{y}_1) - \mathbf{x}\|^2$, $b_{\mathbf{x}, \mathbf{y}_2} = \frac{1}{n}\|\mathbf{y}_2 - \mathbf{x}\|^2$ and $c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2} = -\frac{2}{n}(f(\mathbf{y}_1) - \mathbf{x})^\top(\mathbf{y}_2 - \mathbf{x})$ we have

$$\begin{aligned}\mathbb{V}_{\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}}\left\{\frac{1}{n}\|f(\mathbf{y}_1) - \mathbf{y}_2\|^2\right\} &= \mathbb{V}_{\mathbf{y}_1, \mathbf{y}_2, \mathbf{x}}\{a_{\mathbf{x}, \mathbf{y}_1} + b_{\mathbf{x}, \mathbf{y}_2} + c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2}\} \\ &= \mathbb{V}\{a_{\mathbf{x}, \mathbf{y}_1}\} + \Delta\end{aligned}$$

with

$$\begin{aligned}\Delta &= \mathbb{V}\{b_{\mathbf{x}, \mathbf{y}_2}\} + \mathbb{V}\{c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2}\} \\ &\quad + 2\mathbb{E}_{\mathbf{x}}\{\text{Cov}\{a_{\mathbf{x}, \mathbf{y}_1}, b_{\mathbf{x}, \mathbf{y}_2}\} + \text{Cov}\{a_{\mathbf{x}, \mathbf{y}_1}, c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2}\} + \text{Cov}\{b_{\mathbf{x}, \mathbf{y}_2}, c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2}\}\}\}.\end{aligned}$$

where $\text{Cov}\{\cdot, \cdot\}$ denotes the covariance between two one-dimensional random variables with respect to $p(\mathbf{y}_1, \mathbf{y}_2|\mathbf{x})$.

The first variance term is simply $\mathbb{V}\{b_{\mathbf{x}, \mathbf{y}_2}\} = \mathbb{V}_{\mathbf{y}_2, \mathbf{x}}\{\frac{1}{n}\|\mathbf{y}_2 - \mathbf{x}\|^2\}$, and the second variance can be computed as

$$\begin{aligned}\mathbb{V}_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2}\{c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2}\} &= \frac{4}{n^2}\mathbb{E}_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2}\left\{\left((f(\mathbf{y}_1) - \mathbf{x})^\top(\mathbf{y}_2 - \mathbf{x})\right)^2\right\} \\ &= \frac{4}{n^2}\mathbb{E}_{\mathbf{x}}\left\{\mathbb{E}_{\mathbf{y}_1|\mathbf{x}}\{(f(\mathbf{y}_1) - \mathbf{x})(f(\mathbf{y}_1) - \mathbf{x})^\top\}\mathbb{E}_{\mathbf{y}_2|\mathbf{x}}\{(\mathbf{y}_2 - \mathbf{x})(\mathbf{y}_2 - \mathbf{x})^\top\}\right\} \\ &= \frac{4}{n^2}\mathbb{E}_{\mathbf{x}}\{\text{trace}(\boldsymbol{\Psi}_{\mathbf{x}}\boldsymbol{\Sigma}_{\mathbf{x}})\}\end{aligned}$$

where the first line uses the fact that $\mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2|\mathbf{x}}\{c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2}\} = 0$, and the third line uses the definitions $\boldsymbol{\Sigma}_{\mathbf{x}} = \mathbb{E}_{\mathbf{y}_2|\mathbf{x}}\{(\mathbf{y}_2 - \mathbf{x})(\mathbf{y}_2 - \mathbf{x})^\top\}$ and $\boldsymbol{\Psi}_{\mathbf{x}} = \mathbb{E}_{\mathbf{y}_1|\mathbf{x}}\{(f(\mathbf{y}_1) - \mathbf{x})(f(\mathbf{y}_1) - \mathbf{x})^\top\}$.

We have that $\text{Cov}\{a_{\mathbf{x}, \mathbf{y}_1}, b_{\mathbf{x}, \mathbf{y}_2}\} = 0$ since $a_{\mathbf{x}, \mathbf{y}_1}$ and $b_{\mathbf{x}, \mathbf{y}_2}$ are independent conditioned on \mathbf{x} . We also have that

$$\begin{aligned}\text{Cov}\{a_{\mathbf{x}, \mathbf{y}_1}, c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2}\} &= \mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2|\mathbf{x}}\{(a_{\mathbf{x}, \mathbf{y}_1} - \mathbb{E}_{\mathbf{y}_1|\mathbf{x}}\{a_{\mathbf{x}, \mathbf{y}_1}\})c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2}\} \\ &= \mathbb{E}_{\mathbf{y}_1|\mathbf{x}}\{(a_{\mathbf{x}, \mathbf{y}_1} - \mathbb{E}_{\mathbf{y}_1|\mathbf{x}}\{a_{\mathbf{x}, \mathbf{y}_1}\})\mathbb{E}_{\mathbf{y}_2|\mathbf{x}}\{c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2}\}\} \\ &= 0\end{aligned}$$

The remaining covariance term can be computed as

$$\begin{aligned}
& \text{Cov} \{ b_{\mathbf{x}, \mathbf{y}_2}, c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2} \} \\
&= \mathbb{E}_{\mathbf{y}_2 | \mathbf{x}} \left\{ \left(b_{\mathbf{x}, \mathbf{y}_2} - \mathbb{E}_{\mathbf{y}_2 | \mathbf{x}} \{ b_{\mathbf{x}, \mathbf{y}_2} \} \right) \mathbb{E}_{\mathbf{y}_1 | \mathbf{x}} \{ c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2} \} \right\} \\
&= \mathbb{E}_{\mathbf{y}_2 | \mathbf{x}} \{ b_{\mathbf{x}, \mathbf{y}_2} \mathbb{E}_{\mathbf{y}_1 | \mathbf{x}} \{ c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2} \} \} - \mathbb{E}_{\mathbf{y}_2 | \mathbf{x}} \{ b_{\mathbf{x}, \mathbf{y}_2} \} \mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2 | \mathbf{x}} \{ c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2} \} \\
&= \mathbb{E}_{\mathbf{y}_2 | \mathbf{x}} \{ b_{\mathbf{x}, \mathbf{y}_2} \mathbb{E}_{\mathbf{y}_1 | \mathbf{x}} \{ c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2} \} \} \\
&= -\frac{2}{n^2} \mathbb{E}_{\mathbf{y}_2 | \mathbf{x}} \left\{ \|\mathbf{y}_2 - \mathbf{x}\|^2 (\mathbf{y}_2 - \mathbf{x})^\top \right\} \left(\mathbb{E}_{\mathbf{y}_1 | \mathbf{x}} \{ f(\mathbf{y}_1) \} - \mathbf{x} \right)
\end{aligned}$$

where the second and third lines use the fact that $\mathbb{E}_{\mathbf{y}_1, \mathbf{y}_2 | \mathbf{x}} \{ c_{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2} \} = 0$. \square

Bibliography

- [1] J. A. Fessler, “Model-based image reconstruction for MRI,” *IEEE Signal Processing Magazine*, vol. 27, no. 4, pp. 81–89, 2010.
- [2] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” vol. 60, no. 1, pp. 259–268.
- [3] J. Zbontar, F. Knoll, A. Sriram, T. Murrell, Z. Huang, M. J. Muckley, A. Defazio, R. Stern, P. Johnson, M. Bruno, M. Parente, K. J. Geras, J. Katsnelson, H. Chandarana, Z. Zhang, M. Drozdzal, A. Romero, M. Rabbat, P. Vincent, N. Yakubova, J. Pinkerton, D. Wang, E. Owens, C. L. Zitnick, M. P. Recht, D. K. Sodickson, and Y. W. Lui, “fastMRI: An Open Dataset and Benchmarks for Accelerated MRI,” Dec. 2019.
- [4] E. Shimron, J. I. Tamir, K. Wang, and M. Lustig, “Implicit data crimes: Machine learning bias arising from misuse of public data,” *Proceedings of the National Academy of Sciences*, vol. 119, no. 13, p. e2117203119, 2022.
- [5] F. Luisier, C. Vonesch, T. Blu, and M. Unser, “Fast interscale wavelet denoising of poisson-corrupted images,” *Signal Processing*, vol. 90, no. 2, pp. 415–427, 2010. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0165168409003016>
- [6] S. A. Hussein, T. Tirer, and R. Giryes, “Correction Filter for Single Image Super-Resolution: Robustifying Off-the-Shelf Deep Super-Resolvers,” May 2020.
- [7] J. Dong, L. Valzania, A. Maillard, T.-a. Pham, S. Gigan, and M. Unser, “Phase Retrieval: From Computational Imaging to Machine Learning,” *IEEE Signal Processing Magazine*, vol. 40, no. 1, pp. 45–57, Jan. 2023.
- [8] E. Soubies, T.-A. Pham, and M. Unser, “Efficient inversion of multiple-scattering model for optical diffraction tomography,” *Optics express*, vol. 25, no. 18, pp. 21 786–21 800, 2017.
- [9] E. Candes and M. Wakin, “An Introduction To Compressive Sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, Mar. 2008.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

- [11] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, “Deep Convolutional Neural Network for Inverse Problems in Imaging,” *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, Sep. 2017.
- [12] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.
- [13] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, “SwinIR: Image restoration using swin transformer,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 1833–1844.
- [14] J. Adler and O. Öktem, “Learned primal-dual reconstruction,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1322–1332, 2018.
- [15] K. Hammernik, T. Klatzer, E. Kobler, M. P. Recht, D. K. Sodickson, T. Pock, and F. Knoll, “Learning a variational network for reconstruction of accelerated mri data,” *Magnetic Resonance in Medicine*, vol. 79, no. 6, pp. 3055–3071, 2018. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/mrm.26977>
- [16] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, “Do imagenet classifiers generalize to imagenet?” in *International Conference on Machine Learning*, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:67855879>
- [17] M. Terris, S. Hurault, M. Song, and J. Tachella, Reconstruct Anything Model: A lightweight foundation model for computational imaging.
- [18] C. Belthangady and L. A. Royer, “Applications, Promises, and Pitfalls of Deep Learning for Fluorescence Image Reconstruction,” *Nature Methods*, vol. 16, pp. 1215–1225, Feb. 2019.
- [19] M. M. Kashyap, A. Tambwekar, K. Manohara, and S. Natarajan, “Speech Denoising Without Clean Training Data: A Noise2Noise Approach,” in *Proc. Interspeech 2021*, 2021, pp. 2716–2720.
- [20] P. H. Casajus, T. Ritschel, and T. Ropinski, “Total Denoising: Unsupervised Learning of 3D Point Cloud Cleaning,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 52–60.
- [21] A. Krull, T.-O. Buchholz, and F. Jug, “Noise2Void - Learning Denoising From Single Noisy Images,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, Jun. 2019, pp. 2124–2132.
- [22] A. Krull, T. Vičar, M. Prakash, M. Lalit, and F. Jug, “Probabilistic Noise2Void: Unsupervised Content-Aware Denoising,” *Frontiers in Computer Science*, vol. 2, p. 5, Feb. 2020.
- [23] S. Laine, T. Karras, J. Lehtinen, and T. Aila, “High-Quality Self-Supervised Deep Image Denoising,” in *Advances in Neural Information Processing Systems*, 2019.

- [24] E. Dalsasso, L. Denis, and F. Tupin, “SAR2SAR: A semi-supervised despeckling algorithm for SAR images,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 4321–4329, 2021.
- [25] A. A. Hendriksen, D. M. Pelt, and K. J. Batenburg, “Noise2Inverse: Self-supervised deep convolutional denoising for tomography,” *IEEE Transactions on Computational Imaging*, vol. 6, pp. 1320–1335, 2020.
- [26] D. Chen, J. Tachella, and M. E. Davies, “Equivariant Imaging: Learning Beyond the Range Space,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4379–4388.
- [27] B. Yaman, S. A. H. Hosseini, S. Moeller, J. Ellermann, K. Uğurbil, and M. Akçakaya, “Self-Supervised Physics-Based Deep Learning MRI Reconstruction Without Fully-Sampled Data,” in *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*, Apr. 2020, pp. 921–925.
- [28] J. Tachella, M. Davies, and L. Jacques, “UNSURE: self-supervised learning with unknown noise level and stein’s unbiased risk estimate,” in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: <https://openreview.net/forum?id=ScVnYBaSEw>
- [29] M. Raphan and E. P. Simoncelli, “Least Squares Estimation Without Priors or Supervision,” *Neural Computation*, vol. 23, no. 2, pp. 374–420, Feb. 2011.
- [30] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales, “Self-Supervised Representation Learning: Introduction, Advances and Challenges,” *IEEE Signal Processing Magazine*, vol. 39, no. 3, pp. 42–62, May 2022.
- [31] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A Simple Framework for Contrastive Learning of Visual Representations,” in *Proceedings of the 37th International Conference on Machine Learning*. PMLR, Nov. 2020, pp. 1597–1607.
- [32] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, “Bootstrap your own latent: A new approach to self-supervised Learning,” Sep. 2020.
- [33] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, “Dinov2: Learning robust visual features without supervision,” *arXiv preprint arXiv:2304.07193*, 2023.
- [34] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked Autoencoders Are Scalable Vision Learners,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 000–16 009.
- [35] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Deep Image Prior,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

- [36] R. Heckel and P. Hand, “Deep decoder: Concise image representations from untrained non-convolutional networks,” in *International Conference on Learning Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=rylV-2C9KQ>
- [37] M. Z. Darestani and R. Heckel, “Accelerated mri with un-trained neural networks,” *IEEE Transactions on Computational Imaging*, vol. 7, pp. 724–733, 2021.
- [38] D. Ren, K. Zhang, Q. Wang, Q. Hu, and W. Zuo, “Neural Blind Deconvolution Using Deep Priors,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2020, pp. 3338–3347.
- [39] J. Tachella, J. Tang, and M. Davies, “The neural tangent link between cnn denoisers and non-local filters,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8618–8627.
- [40] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, “Noise2Noise: Learning Image Restoration without Clean Data,” in *Proceedings of the 35th International Conference on Machine Learning*. PMLR, Oct. 2018.
- [41] G. Daras, H. Chung, C.-H. Lai, Y. Mitsufuji, J. C. Ye, P. Milanfar, A. G. Dimakis, and M. Delbracio, “A Survey on Diffusion Models for Inverse Problems,” Sep. 2024.
- [42] U. S. Kamilov, C. A. Bouman, G. T. Buzzard, and B. Wohlberg, “Plug-and-Play Methods for Integrating Physical and Learned Models in Computational Imaging: Theory, algorithms, and applications,” vol. 40, no. 1, pp. 85–97.
- [43] F. Rozet, G. Andry, F. Lanusse, and G. Louppe, “Learning Diffusion Priors from Observations by Expectation Maximization,” vol. 37, pp. 87647–87682. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2024/hash/9f94298bac4668db4dc77ddb0a244301-Abstract-Conference.html
- [44] G. Daras, A. Dimakis, and C. C. Daskalakis, “Consistent Diffusion Meets Tweedie: Training Exact Ambient Diffusion Models with Noisy Data,” in *Proceedings of the 41st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, and F. Berkenkamp, Eds., vol. 235. PMLR, Jul. 2024, pp. 10 091–10 108. [Online]. Available: <https://proceedings.mlr.press/v235/daras24a.html>
- [45] J. Tachella, M. Terris, S. Hurault, A. Wang, D. Chen, M.-H. Nguyen, M. Song, T. Davies, L. Davy, J. Dong, P. Escande, J. Hertrich, Z. Hu, T. I. Liaudat, N. Laurent, B. Levac, M. Massias, T. Moreau, T. Modrzyk, B. Monroy, S. Neumayer, J. Scanvic, F. Sarron, V. Sechaud, G. Schramm, R. Vo, and P. Weiss, “Deepinverse: A python package for solving imaging inverse problems with deep learning,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.20160>
- [46] C. L. Mallows, “Some Comments on CP,” *Technometrics*, vol. 15, no. 4, pp. 661–675, 1973.

- [47] B. Efron, “The Estimation of Prediction Error: Covariance Penalties and Cross-Validation,” *Journal of the American Statistical Association*, vol. 99, no. 467, pp. 619–632, Sep. 2004.
- [48] T. Bepler, K. Kelley, A. J. Noble, and B. Berger, “Topaz-Denoise: General deep denoising models for cryoEM and cryoET,” *Nature Communications*, vol. 11, no. 1, p. 5208, Oct. 2020.
- [49] E. Dalsasso, L. Denis, and F. Tupin, “As if by magic: Self-supervised training of deep despeckling networks with MERLIN,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–13, 2022.
- [50] T. Ehret, A. Davy, J.-M. Morel, G. Facciolo, and P. Arias, “Model-blind video denoising via frame-to-frame training,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 361–11 370.
- [51] N. Moran, D. Schmidt, Y. Zhong, and P. Coady, “Noisier2Noise: Learning to Denoise From Unpaired Noisy Data,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, Jun. 2020, pp. 12 061–12 069.
- [52] T. Pang, H. Zheng, Y. Quan, and H. Ji, “Recorrupted-to-Recorrupted: Unsupervised Deep Learning for Image Denoising,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2043–2052.
- [53] C. M. Stein, “Estimation of the Mean of a Multivariate Normal Distribution,” *The Annals of Statistics*, vol. 9, no. 6, pp. 1135–1151, 1981.
- [54] N. L. Oliveira, J. Lei, and R. J. Tibshirani, “Unbiased Risk Estimation in the Normal Means Problem via Coupled Bootstrap Techniques,” Oct. 2022.
- [55] B. Monroy, J. Bacca, and J. Tachella, “Generalized Recorrupted-to-Recorrupted: Self-Supervised Learning Beyond Gaussian Noise,” in *2025 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2024-12-05.
- [56] J. Rapp, J. Tachella, Y. Altmann, S. McLaughlin, and V. K. Goyal, “Advances in Single-Photon Lidar for Autonomous Vehicles: Working Principles, Challenges, and Recent Advances,” *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 62–71, Jul. 2020.
- [57] B. Efron, *Exponential families in theory and practice*. Cambridge University Press, 2022.
- [58] J. Leiner, B. Duan, L. Wasserman, and A. Ramdas, “Data fission: splitting a single data point,” *Journal of the American Statistical Association*, vol. 120, no. 549, pp. 135–146, 2025.
- [59] D. L. Donoho and I. M. Johnstone, “Adapting to unknown smoothness via wavelet shrinkage,” *Journal of the american statistical association*, vol. 90, no. 432, pp. 1200–1224, 1995.

- [60] C. A. Metzler, A. Mousavi, R. Heckel, and R. G. Baraniuk, “Unsupervised Learning with Stein’s Unbiased Risk Estimator,” Jul. 2016.
- [61] M. Zhussip, S. Soltanayev, and S. Y. Chun, “Extending Stein’ s unbiased risk estimator to train deep denoisers with correlated pairs of noisy images,” in *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc., 2019.
- [62] D. Chen, J. Tachella, and M. E. Davies, “Robust Equivariant Imaging: A Fully Unsupervised Framework for Learning To Image From Noisy and Partial Measurements,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5647–5656.
- [63] H. M. Hudson, “A Natural Identity for Exponential Families with Applications in Multiparameter Estimation,” *The Annals of Statistics*, vol. 6, no. 3, pp. 473–484, 1978.
- [64] Y. Le Montagner, E. D. Angelini, and J.-C. Olivo-Marin, “An Unbiased Risk Estimator for Image Denoising in the Presence of Mixed Poisson–Gaussian Noise,” *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1255–1268, Mar. 2014.
- [65] Y. Zhang, Y. Zhu, E. Nichols, Q. Wang, S. Zhang, C. Smith, and S. Howard, “A Poisson-Gaussian Denoising Dataset with Real Fluorescence Microscopy Images,” Apr. 2019.
- [66] Q. Ding, Y. Long, X. Zhang, and J. A. Fessler, “Statistical Image Reconstruction Using Mixed Poisson-Gaussian Noise Model for X-Ray CT,” Jan. 2018.
- [67] Y. Eldar, “Generalized SURE for Exponential Families: Applications to Regularization,” *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 471–481, Feb. 2009.
- [68] S. Soltanayev, R. Giryes, S. Y. Chun, and Y. C. Eldar, “On divergence approximations for unsupervised training of deep denoisers based on stein’s unbiased risk estimator,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 3592–3596.
- [69] M. F. Hutchinson, “A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines,” *Communications in Statistics-Simulation and Computation*, vol. 18, no. 3, pp. 1059–1076, 1989.
- [70] S. Ramani, T. Blu, and M. Unser, “Monte-Carlo Sure: A Black-Box Optimization of Regularization Parameters for General Denoising Algorithms,” *IEEE Transactions on Image Processing*, vol. 17, no. 9, pp. 1540–1554, Sep. 2008.
- [71] N. L. Oliveira, J. Lei, and R. J. Tibshirani, “Unbiased Test Error Estimation in the Poisson Means Problem via Coupled Bootstrap Techniques,” Aug. 2023.
- [72] K. Kim and J. C. Ye, “Noise2Score: Tweedie’s Approach to Self-Supervised Image Denoising without Clean Images,” in *Advances in Neural Information Processing Systems*, 2021.

- [73] K. Kim, T. Kwon, and J. C. Ye, “Noise distribution adaptive self-supervised image denoising using tweedie distribution and score matching,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2008–2016. [Online]. Available: http://openaccess.thecvf.com/content/CVPR2022/html/Kim_Noise_Distribution_Adaptive_Self-Supervised_Image_Denoising_Using_Tweedie_Distribution_and_CVPR_2022_paper.html
- [74] R. Gribonval, “Should Penalized Least Squares Regression be Interpreted as Maximum A Posteriori Estimation?” *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2405–2410, May 2011.
- [75] J. Batson and L. Royer, “Noise2Self: Blind Denoising by Self-Supervision,” in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, Jun. 2019.
- [76] T. Huang, S. Li, X. Jia, H. Lu, and J. Liu, “Neighbor2Neighbor: Self-Supervised Denoising from Single Noisy Images,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Nashville, TN, USA: IEEE, Jun. 2021, pp. 14 776–14 785.
- [77] C. Millard and M. Chiew, “A theoretical framework for self-supervised MR image reconstruction using sub-sampling via variable density Noisier2Noise,” *IEEE transactions on computational imaging*, 2023.
- [78] H. Robbins, “The empirical Bayes approach to statistical decision problems,” *The Annals of Mathematical Statistics*, vol. 35, no. 1, pp. 1–20, 1964.
- [79] J. Tachella, D. Chen, and M. Davies, “Sensing Theorems for Unsupervised Learning in Linear Inverse Problems,” *Journal of Machine Learning Research (JMLR)*, 2023.
- [80] M. Prakash, A. Krull, and F. Jug, “Fully Unsupervised Diversity Denoising with Convolutional Variational Autoencoders,” in *Proceedings of the International Conference on Learning Representations*. [Online]. Available: <https://openreview.net/forum?id=agHLCOBM5jP>
- [81] M. Prakash, M. Delbracio, P. Milanfar, and F. Jug, “Interpretable Unsupervised Diversity Denoising and Artefact Removal,” in *Proceedings of the International Conference on Learning Representations*, Oct. 2021.
- [82] A. Bora, E. Price, and A. G. Dimakis, “AmbientGAN: Generative Models From Lossy Measurements,” in *International Conference on Learning Representations*, 2018.
- [83] B. Kawar, N. Elata, T. Michaeli, and M. Elad, “GSURE-based diffusion model training with corrupted data,” *arXiv preprint arXiv:2305.13128*, 2023.
- [84] Z. Wang, J. Liu, G. Li, and H. Han, “Blind2Unblind: Self-Supervised Image Denoising With Visible Blind Spots,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2027–2036.

- [85] G. Daras, K. Shah, Y. Dagan, A. Gollakota, A. Dimakis, and A. Klivans, “Ambient Diffusion: Learning Clean Distributions from Corrupted Data,” *Advances in Neural Information Processing Systems*, vol. 36, Feb. 2024.
- [86] W. Gan, C. Ying, P. E. Boroojeni, T. Wang, C. Eldeniz, Y. Hu, J. Liu, Y. Chen, H. An, and U. S. Kamilov, “Self-supervised deep equilibrium models with theoretical guarantees and applications to mri reconstruction,” *IEEE Transactions on Computational Imaging*, vol. 9, pp. 796–807, 2023.
- [87] J. Tachella, D. Chen, and M. Davies, “Unsupervised Learning From Incomplete Measurements for Inverse Problems,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 4983–4995, Dec. 2022.
- [88] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, *Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges*. arXiv, May 2021.
- [89] T. S. Cohen and M. Welling, “Steerable CNNs,” in *International Conference on Learning Representations*, 2017. [Online]. Available: <https://openreview.net/forum?id=rJQKYt5ll>
- [90] V. Sechaud, J. Scanvic, Q. Barthélémy, P. Abry, and J. Tachella, “Equivariant splitting: Self-supervised learning from incomplete data,” *arXiv preprint arXiv:2510.00929*, 2025.
- [91] J. Scanvic, M. Davies, P. Abry, and J. Tachella, “Self-Supervised Learning for Image Super-Resolution and Deblurring,” Dec. 2023.
- [92] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved Training of Wasserstein GANs. [Online]. Available: <http://arxiv.org/abs/1704.00028>
- [93] E. K. Cole, F. Ong, S. S. Vasanawala, and J. M. Pauly, “Fast Unsupervised MRI Reconstruction Without Fully-Sampled Ground Truth Data Using Generative Adversarial Networks,” in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. Montreal, BC, Canada: IEEE, Oct. 2021, pp. 3971–3980.
- [94] M. Bendel, R. Ahmad, and P. Schniter, “A regularized conditional gan for posterior sampling in image recovery problems,” *Advances in neural information processing systems*, vol. 36, pp. 68 673–68 684, 2023.
- [95] A. Aali, G. Daras, B. Levac, S. Kumar, A. Dimakis, and J. Tamir, “Ambient diffusion posterior sampling: Solving inverse problems with diffusion models trained on corrupted data,” in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: <https://openreview.net/forum?id=qeXcMutEZY>
- [96] H. Cramer and H. Wold, “Some Theorems on Distribution Functions,” *Journal of the London Mathematical Society.*, vol. 11, no. 4, pp. 290–294, 1936.
- [97] C. Fefferman, “Reconstructing a neural net from its output,” in *Advances in neural information processing systems*, 1993.
- [98] K. Falconer, *Fractal geometry: mathematical foundations and applications*. John Wiley & Sons, 2013.

- [99] G. Puy, M. E. Davies, and R. Gribonval, “Recipes for Stable Linear Embeddings From Hilbert Spaces to \mathbb{R}^m ,” *IEEE Transactions on Information Theory*, vol. 63, no. 4, pp. 2171–2187, Apr. 2017.
- [100] T. Sauer, J. A. Yorke, and M. Casdagli, “Embedology,” *Journal of Statistical Physics*, vol. 65, no. 3, pp. 579–616, Nov. 1991.
- [101] A. Wang and M. Davies, “Perspective-Equivariant Imaging: An Unsupervised Framework for Multispectral Pansharpening,” Mar. 2024.
- [102] M. Hardt and B. Recht, *Patterns, predictions, and actions: Foundations of machine learning*. Princeton University Press, 2022.
- [103] T. Klug, D. Atik, and R. Heckel, “Analyzing the Sample Complexity of Self-Supervised Image Reconstruction Methods,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 65 869–65 893, Dec. 2023.
- [104] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, “Plug-and-play image restoration with deep denoiser prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 6360–6376, 2021.
- [105] P. C. Bellec and C.-H. Zhang, “Second-order Stein: SURE for SURE and other applications in high-dimensional inference,” *The Annals of Statistics*, vol. 49, no. 4, pp. 1864–1903, 2021.
- [106] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning (still) requires rethinking generalization,” *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.
- [107] M. Hardt, B. Recht, and Y. Singer, “Train faster, generalize better: Stability of stochastic gradient descent,” in *International conference on machine learning*. PMLR, 2016, pp. 1225–1234.
- [108] S. Mohan, J. L. Vincent, R. Manzorro, P. Crozier, C. Fernandez-Granda, and E. Simoncelli, “Adaptive denoising via gaintuning,” *Advances in neural information processing systems*, vol. 34, pp. 23 727–23 740, 2021.
- [109] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk, “Robust 1-Bit Compressive Sensing via Binary Stable Embeddings of Sparse Vectors,” Nov. 2015.
- [110] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, “Phase Retrieval with Application to Optical Imaging: A contemporary overview,” *IEEE Signal Processing Magazine*, vol. 32, no. 3, pp. 87–109, May 2015.
- [111] J. Tachella and L. Jacques, “Learning to reconstruct signals from binary measurements alone,” *Transactions on Machine Learning Research*, 2023, featured Certification. [Online]. Available: <https://openreview.net/forum?id=ioFIAQOBOS>
- [112] V. Sechaud, L. Jacques, P. Abry, and J. Tachella, “Equivariance-based self-supervised learning for audio signal recovery from clipped measurements,” in *EUSIPCO 2024*, Aug. 2024.

- [113] Y. Hu, W. Gan, C. Ying, T. Wang, C. Eldeniz, J. Liu, Y. Chen, H. An, and U. S. Kamilov, “SPICE: Self-Supervised Learning for MRI with Automatic Coil Sensitivity Estimation,” Oct. 2022.
- [114] M. Kellman, K. Zhang, E. Markley, J. Tamir, E. Bostan, M. Lustig, and L. Waller, “Memory-efficient learning for large-scale computational imaging,” *IEEE Transactions on Computational Imaging*, vol. 6, pp. 1403–1414, 2020.
- [115] F. Ong, X. Zhu, J. Y. Cheng, K. M. Johnson, P. E. Z. Larson, S. S. Vasanawala, and M. Lustig, “Extreme MRI: Large-scale volumetric dynamic imaging from continuous non-gated acquisitions,” *Magnetic Resonance in Medicine*, vol. 84, no. 4, pp. 1763–1780, Oct. 2020.
- [116] J. Rudzusika, B. Bajić, T. Koehler, and O. Öktem, “3D Helical CT Reconstruction With a Memory Efficient Learned Primal-Dual Architecture,” *IEEE Transactions on Computational Imaging*, vol. 10, pp. 1414–1424, 2024.
- [117] C. Ophus, “Four-dimensional scanning transmission electron microscopy (4d-stem): From scanning nanodiffraction to ptychography and beyond,” *Microscopy and Microanalysis*, vol. 25, no. 3, pp. 563–582, 06 2019. [Online]. Available: <https://doi.org/10.1017/S1431927619000497>
- [118] H. Erdogan and J. A. Fessler, “Ordered subsets algorithms for transmission tomography,” *Physics in Medicine & Biology*, vol. 44, no. 11, p. 2835, Nov. 1999. [Online]. Available: <https://dx.doi.org/10.1088/0031-9155/44/11/311>
- [119] J. Tang, K. Egiazarian, M. Golbabaei, and M. Davies, “The practicality of stochastic optimization in imaging inverse problems,” *IEEE Transactions on Computational Imaging*, vol. 6, pp. 1471–1485, 2020.
- [120] O. Kosomaa, S. Laine, T. Karras, M. Aittala, and J. Lehtinen, “Simulator-Based Self-Supervision for Learned 3D Tomography Reconstruction,” May 2023.
- [121] J. Tang, S. Mukherjee, and C.-B. Schönlieb, “Stochastic primal-dual deep unrolling,” 2022. [Online]. Available: <https://arxiv.org/abs/2110.10093>
- [122] Z. Xia and A. Chakrabarti, “Training image estimators without image ground truth,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [123] H. Manor and T. Michaeli, “On the Posterior Distribution in Denoising: Application to Uncertainty Quantification,” in *The Twelfth International Conference on Learning Representations*, 2024.
- [124] J. Tachella and M. Pereyra, “Equivariant Bootstrapping for Uncertainty Quantification in Imaging Inverse Problems,” in *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*. PMLR, Apr. 2024.
- [125] D. P. Kingma and M. Welling, “Auto-Encoding Variational Bayes,” in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

- [126] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, 2014, pp. 2672–2680. [Online]. Available: <http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf>
- [127] D. Donoho, “High-dimensional data analysis: The curses and blessings of dimensionality,” *AMS Math Challenges Lecture*, pp. 1–32, 01 2000.
- [128] A. Wang, S. McDonagh, and M. Davies, “Benchmarking Self-Supervised Learning Methods for Accelerated MRI Reconstruction,” 2025. [Online]. Available: <https://arxiv.org/abs/2502.14009>