CURE: Curvature Regularization For Missing Data Recovery

Bin Dong *, Haocheng Ju†, Yiping Lu‡, and Zuoqiang Shi§

Abstract. Missing data recovery is an important and yet challenging problem in imaging and data science. Successful models often adopt certain carefully chosen regularization. Recently, the low dimensional manifold model (LDMM) was introduced by [37] and shown effective in image inpainting. The authors of [37] observed that enforcing low dimensionality on image patch manifold serves as a good image regularizer. In this paper, we observe that having only the low dimensional manifold regularization is not enough sometimes, and we need smoothness of the manifold as well. For that, we introduce a new regularization by combining the low dimensional manifold regularization with a higher order **CU**rvature **RE**gularization, and we call this new regularization CURE for short. The key step of CURE is to solve a biharmonic equation on a manifold. We further introduce a weighted version of CURE, called WeCURE, in a similar manner as the weighted nonlocal Laplacian (WNLL) method [44]. Numerical experiments for image inpainting and semi-supervised learning show that the proposed CURE and WeCURE significantly outperform LDMM and WNLL respectively.

Key words. Graph Laplacian, Nonlocal Methods, Point Cloud, Biharmonic Equation, Interpolation, Image Inpainting.

AMS subject classifications. 62H35 65D18 68U10 58C40 58J50

1. Introduction.

Missing data recovery is a fundamental problem in imaging science and data analysis. In many cases, it can be formulated as a function interpolation problem in multiple dimension spaces. Let $u : \mathbb{R}^d \to \mathbb{R}$ be an unknown function. We would like to acquire its values on a set of points $P = \{\boldsymbol{p}_1, \dots, \boldsymbol{p}_n\} \subset \mathbb{R}^d$. However, due to practical limitations, we are only able to observe its values on a subset $S = \{\boldsymbol{s}_1, \dots, \boldsymbol{s}_m\} \subset P$. The goal of missing data recovery is to reconstruct the missing values of u based on the observed values in S. In this paper, we focus on two kinds of typical and important tasks of missing data recovery, i.e., semi-supervised learning and image inpainting, though it can be well applied to other related tasks as well.

Since the problem of missing data recovery is an under-determined inverse problem, we can only hope to recover the missing values of u if we have certain prior knowledge on u, e.g., u belonging to a certain function class or having certain mathematical or statistical properties. Successful models include Rudin–Osher–Fatemi (ROF) model [41] and its variants [4,13,26,38], the applied harmonic analysis models such as wavelets [18, 47], curvelet [46], shearlet [22, 32]

*Beijing International Center for Mathematical Research, Peking University, Beijing, 100871 China.(dongbin@math.pku.edu.cn)

†School Of Mathematical Science, Peking University, Beijing, 100871 China.(jhc9912@pku.edu.cn)

‡Institute for Computational and Mathematical Engineering (ICME), Stanford University, Stanford, CA, 94305.(yplu@stanford.edu)

§Department of Mathematical Sciences, Tsinghua University, Beijing, 100084 China.(zqshi@tsinghua.edu.cn)

and wavelet frame [2,8,9,12,20,50], the Bayesian statistics based methods [40,42,51]; and the list goes on.

More recently, people started to use low dimensional manifolds to describe the underlying relationship between the data points which serves as an effective geometric prior on the interpolant. For example, [37, 39] observed that image patches, regarded as data points in a high dimension space, often lie on a low dimensional manifold; and [15, 52] allowed the data lie close to (but may not be on) a certain low dimensional manifold.

To harvest the low dimensional property of data, [37] applied the following Dirichlet energy [53] to regularize the dimension of the embedded manifold $\mathcal{M}$

$$\mathrm{LDMM}(u) = \frac{1}{2} \|\nabla_{\mathcal{M}} u\|^2_{L^2(\mathcal{M})} . \tag{1.1}$$

In [37], the authors gave a geometric interpretation of the Dirichlet regularizer. They showed that the dimension of a smooth manifold embedded in $\mathbb{R}^d$ can be calculated by a simple formula

$$dim(\mathcal{M})(\boldsymbol{x}) = \sum_{j=1}^{d} |\nabla_{\mathcal{M}} \alpha_i(\boldsymbol{x})|^2$$

where α_i is the coordinate function, for any $\boldsymbol{x} = (x_1, \cdots, x_d) \in \mathcal{M} \subset \mathbb{R}^d$, $\alpha_i(\boldsymbol{x}) = x_i$
This means that we can minimize the Dirichlet energy to enforce a penalty on the (local) dimensions of the underlying manifold. As a result, the authors referred to their method as the low dimensional manifold model (LDMM). To recover missing data, they proposed to minimize the Dirichlet energy subject to the constraints $u(\boldsymbol{s}) = g(\boldsymbol{s})$, $\forall \mathbf{s} \in S$, where $g : S \to \mathbb{R}$ denotes the observed part of the underlying function u.

1.1. Higher Order Regularization. Only considering the low dimension structure of the manifold does not readily ensure smoothness of the reconstructed manifold which may lead to unsatisfactory results [11,23,34]. As a simple demonstration, we show in Figure 1 a degenerated interpolation result from the two data points labeled in red. Although the interpolated surface is also a low dimensional manifold, it is certainly not a smooth interpolation.

In this paper, we look for the proper interpolation by not only assuming low dimensionality of the manifold, but also the smoothness. For that, in addition to the Dirichlet energy, we further introduce a **CU**rvature **RE**gularization (**CURE**) term via Laplace−Beltrami operator $\Delta_{\mathcal{M}}$. The proposed CURE energy reads as follows

$$\mathrm{CURE}(u) = \int_{\mathcal{M}} |\nabla_{\mathcal{M}} u(\boldsymbol{x})|^2 d\boldsymbol{x} + \frac{\lambda}{2} \int_{\mathcal{M}} (\Delta_{\mathcal{M}} u(\boldsymbol{x}))^2 d\boldsymbol{x} \tag{1.2}$$

where LDMM is given by (1.1). Note that regularizing the curvature by introducing higher order energy term has already been proposed in image processing [43]. However, to the best of our knowledge, we are the first to promote curvature-like regularization for nonlocal image processing. Furthermore, inspired by the weighted nonlocal Laplacian (WNLL) method proposed by [44] which can preserve the symmetry of the Laplace operator, we propose a

weighted CURE (WeCURE) model which can significantly improve the results over the CURE model. To demonstrate the effectiveness of CURE and WeCURE, we test our model on semi-supervised learning and image inpainting task. Numerical results show that CURE/WeCURE produces significantly better results than LDMM/WNLL in both tasks. A glimpse of the results for image inpainting is shown in Figure 2 where we can see the significant improvement of CURE over LDMM and WeCURE over WNLL. More details and numerical results can be found in section 3 and section 4.

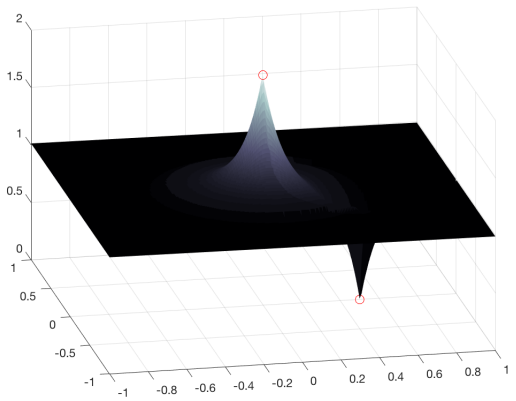


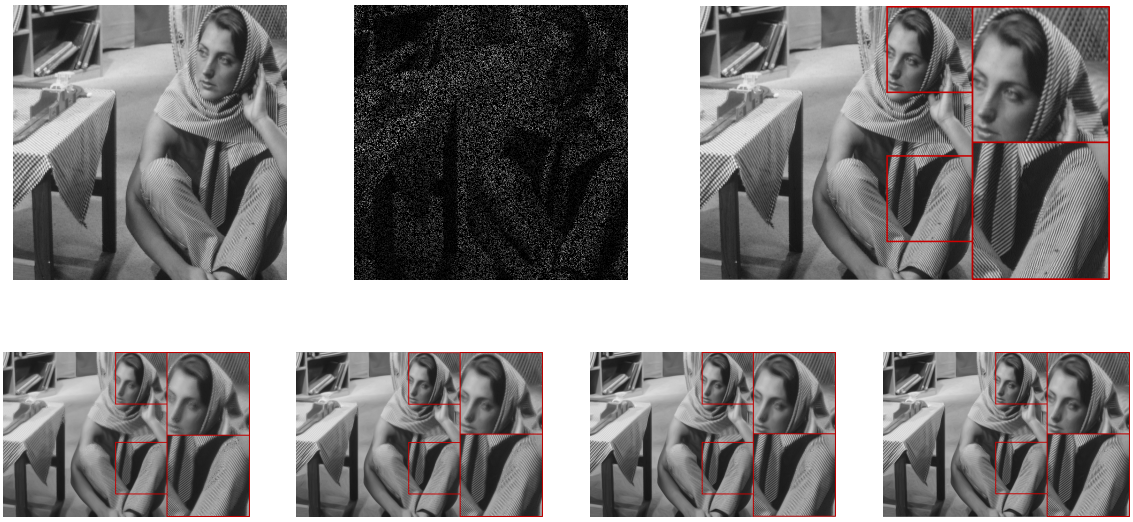
Figure 1: A low dimensional manifold without curvature regularization.

1.2. Other Related Works. Nonlocal patch-based image restoration methods [5, 6, 16, 17, 24, 26, 36] have achieved great success in the literature. In addition, [3, 19, 25] also introduced different graph Laplacian-based regularization on manifold and graphs. Our method, however, focuses on both smoothness and low dimensionality of the underlying data manifold. The most similar work to ours is [1], where the authors also introduced a higher order regularization for semi-supervised learning. The difference is threefold. Firstly, we extend the task to image inpainting rather than just semi-supervised learning. Secondly, we introduce a curvature perspective on the higher order regularization. Last but not least, the newly proposed weighted version of CURE, i.e. WeCURE, has significant performance boost in both image inpainting and semi-supervised learning.

Another approach to regularize the dimension of the manifold is through low-rank matrix completion [27, 28]. The basic idea is to group the patches by similarity and penalized the rank/nuclear norm of the matrix obtained by reshaping the stack of the similar patches. The work in this paper reveals a benefit of PDE-based approaches that higher order information, such as curvature, can be naturally incorporated in the model.

1.3. Organization of the Paper. The paper is organized as follows. The proposed CURE and WeCURE model are introduced in section 2. Numerical comparisons of CURE and WeCURE with LDMM and WNLL for semi-supervised learning and image inpainting are

presented in section 3 and section 4 respectively. The general setting of the asymptotic analysis of the proposed model is presented in Section 5. Conclusions and summary are given in section 6.



LDMM:PSNR=26.81dB, SSIM=0.68

WNLL:PSNR=28.73dB, SSIM=0.73

CURE:PSNR=28.97dB, SSIM=0.75

WeCURE:PSNR=29.78dB, SSIM=0.77

Figure 2: First row: original image, 20% subsampled image, zoom-in views of the original image. Second row: inpainting results of LDMM, WNLL, CURE, WeCURE.

2. Curvature Regularization (CURE): Model and General Algorithm.

In this section, we first propose the CURE model and a weighted version of CURE. Then, we will discuss how (We)CURE can be applied to missing data recovery in general.

2.1. CURE.

Let $\mathcal{M}$ be a smooth manifold embedded in $\mathbb{R}^d$ and locally parameterized as

$$\boldsymbol{x} = \psi(\boldsymbol{\alpha}) : U \subset \mathbb{R}^k \to \mathcal{M} \subset \mathbb{R}^d$$

where $k = dim_{\boldsymbol{x}}(\mathcal{M})$ is the local dimension of $\mathcal{M}$ at $\boldsymbol{x}$, $\boldsymbol{\alpha} = (\alpha^1, \ldots, \alpha^k)^\top \in \mathbb{R}^k$ and $\boldsymbol{x} = (x_1, \ldots, x_d)^\top \in \mathcal{M}$. Let $\boldsymbol{u} = (u_1, u_2, \cdots, u_d)$ be the coordinate function on $\mathcal{M}$, *i.e.* for $\boldsymbol{x} \in \mathcal{M}$

$$u_i(\boldsymbol{x}) = x_i, \ 1 \le i \le d.$$

To enforce smoothness of the underlying manifold, we further regularize the curvature of the manifold. Recall that the mean curvature of a manifold $\mathcal{M}$ is defined as the trace of the second fundamental form divided by dimension k. If the coordinate function $\boldsymbol{u}(\boldsymbol{x})$ is an isometric immersion, the mean curvature can be calculated as $\|\Delta_{\mathcal{M}}\boldsymbol{u}\|_2/k$, where $\Delta_{\mathcal{M}}\boldsymbol{u} = (\Delta_{\mathcal{M}}u_1, \Delta_{\mathcal{M}}u_2, \cdots, \Delta_{\mathcal{M}}u_d)$(see [33] for detail).

Now, we are ready to introduce the CURE energy in continuum setting:

$$\mathrm{CURE}(u) = \int_{\mathcal{M}} |\nabla_{\mathcal{M}} u(\boldsymbol{x})|^2 d\boldsymbol{x} + \frac{\lambda}{2} \int_{\mathcal{M}} (\Delta_{\mathcal{M}} u(\boldsymbol{x}))^2 d\boldsymbol{x},$$

where the first term is the LDMM(u) given in (1.1). The gradient $\nabla_{\mathcal{M}} u$ in LDMM(u) is commonly approximated by the nonlocal gradient in the discrete setting

$$\nabla_{\mathcal{M}} u(\boldsymbol{x}, \boldsymbol{y}) \approx \sqrt{\omega(\boldsymbol{x}, \boldsymbol{y})}(u(\boldsymbol{y}) - u(\boldsymbol{x})) =: \nabla_P u(\boldsymbol{x}, \boldsymbol{y}),\ \boldsymbol{x}, \boldsymbol{y} \in P \subset \mathcal{M},$$

where P is a set with n points on the manifold $\mathcal{M}$. Then,

$$\mathrm{LDMM}(u) \approx \frac{1}{2} \sum_{\boldsymbol{x}, \boldsymbol{y} \in P} w(\boldsymbol{x}, \boldsymbol{y})(u(\boldsymbol{x}) - u(\boldsymbol{y}))^2 := \|\nabla_P u\|_2^2.$$

Here, $w(\boldsymbol{x}, \boldsymbol{y})$ is a given symmetric weight function which is often chosen to be a Gaussian weight $w(\boldsymbol{x}, \boldsymbol{y})$=exp($-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{\sigma^2}$), where σ is a parameter and $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^{\frac{n(n-1)}{2}}$. The negative of the first variation of $\|\nabla_P u\|_2^2$ takes the form

$$-\partial_u \left(\|\nabla_P u\|_2^2 \right) = \sum_{\boldsymbol{y} \in P} w(\boldsymbol{x}, \boldsymbol{y})(u(\boldsymbol{x}) - u(\boldsymbol{y})),$$

which is the nonlocal Laplacian that has been used in image processing [6, 7, 25, 26]. It is also called graph Laplacian in spectral graph and machine learning literature [14, 53]. To simplify the notation, we use GL to denote the graph Laplacian [31, 48, 49]:

$$GLu(\boldsymbol{x}) := \sum_{\boldsymbol{y} \in P} w(\boldsymbol{x}, \boldsymbol{y})(u(\boldsymbol{x}) - u(\boldsymbol{y})).$$

Now, the proposed **CURE** model can be cast as the following optimization problem in the discrete setting

$$\min_u \ \|\nabla_P u\|_2^2 + \frac{\lambda}{2} \|GLu\|_2^2. \tag{2.1}$$

In [44], a weighted nonlocal Laplacian (WNLL) method was introduced to balance the loss at both labeled and unlabeled points and to preserve the symmetry of the Laplace operator at the same time. Let $S \subset P$ be a set with labeled points. The WNLL model in the discrete setting is given by

$$\mathrm{WNLL}(u) = \| (\nabla_P u)_{|P \setminus S} \|_2^2 + \frac{|P|}{|S|} \| (\nabla_P u)_{|S} \|_2^2,$$

where

$$\| (\nabla_P u)_{|S} \|_2^2 := \frac{1}{2} \sum_{\boldsymbol{x} \in S, \boldsymbol{y} \in P} w(\boldsymbol{x}, \boldsymbol{y})(u(\boldsymbol{x}) - u(\boldsymbol{y}))^2,$$

and similarly for $\| (\nabla_P u)_{|P\setminus S} \|_2^2$.

Following a similar idea as that in WNLL, we propose the weighted CURE model (**WeCURE**) in the discrete setting

$$\text{WeCURE}(u) := \text{WNLL}(u) + \lambda \left[\| (GLu)_{|P\setminus S} \|_2^2 + \frac{|P|}{|S|} \| (GLu)_{|S} \|_2^2 \right], \tag{2.2}$$

where

$$\| (GLu)_{|S} \|_2^2 = \sum_{\boldsymbol{x}\in S} \left(\sum_{\boldsymbol{y}\in P} w(\boldsymbol{x},\boldsymbol{y})(u(\boldsymbol{x}) - u(\boldsymbol{y})) \right)^2$$

and similarly for $\| (GLu)_{|P\setminus S} \|_2^2$.

2.2. CURE for Missing Data Recovery. For missing data recovery, we can simply minimize the CURE or WeCURE energy with respect to the constraints $u(\boldsymbol{x}) = g(\boldsymbol{x}), \boldsymbol{x} \in S$ where g is the observed values of the underlying function to be recovered. We discuss implementation details for WeCURE. CURE is a special case of WeCURE with all weights equal to 1.

Recall the definition of the energy function of WeCURE (2.2) and notice that $u(\boldsymbol{x}) = g(\boldsymbol{x}), \boldsymbol{x} \in S$. Then, WeCURE model for missing data recovery can be rewritten as

$$\min_{u|_{P\setminus S}} \text{WNLL}\left(\begin{bmatrix} u|_{P\setminus S} \\ g \end{bmatrix} \right) + \lambda \left\| \sqrt{D} \cdot GL \begin{bmatrix} u|_{P\setminus S} \\ g \end{bmatrix} \right\|_2^2, \tag{2.3}$$

where $D = \text{diag}\{d_1, d_2, \ldots, d_{|P|}\}$ with $d_i = 1$ for $\boldsymbol{x}_i \in P\setminus S$ and $d_i = \frac{|P|}{|S|}$ for $\boldsymbol{x}_i \in S$, and GL is the $|P| \times |P|$ matrix of graph Laplacian. The first variation of (2.3) is

$$\partial_{u|_{P\setminus S}} \text{WeCURE}\left(\begin{bmatrix} u|_{P\setminus S} \\ g \end{bmatrix} \right) = \partial_{u|_{P\setminus S}} \text{WNLL}\left(\begin{bmatrix} u|_{P\setminus S} \\ g \end{bmatrix} \right) + \lambda \partial_{u|_{P\setminus S}} \left\| \sqrt{D} \cdot GL \begin{bmatrix} u|_{P\setminus S} \\ g \end{bmatrix} \right\|_2^2.$$

Note that

$$\left\| \sqrt{D} \cdot GL \begin{bmatrix} u|_{P\setminus S} \\ g \end{bmatrix} \right\|_2^2 = \left\| \sqrt{D} \cdot GL \begin{bmatrix} u|_{P\setminus S} \\ 0 \end{bmatrix} + \sqrt{D} \cdot GL \begin{bmatrix} 0 \\ g \end{bmatrix} \right\|_2^2.$$

Thus

$$\begin{aligned} \partial_{u|_{P\setminus S}} \text{WeCURE}\left(\begin{bmatrix} u|_{P\setminus S} \\ g \end{bmatrix} \right) =& \partial_{u|_{P\setminus S}} \text{WNLL}\left(\begin{bmatrix} u|_{P\setminus S} \\ g \end{bmatrix} \right) \\ &+ 2\lambda GL^T \cdot D \cdot GL \begin{bmatrix} u|_{P\setminus S} \\ 0 \end{bmatrix} + 2\lambda GL^T \cdot D \cdot GL \begin{bmatrix} 0 \\ g \end{bmatrix}. \end{aligned}$$

Then, the solution to problem (2.3) can be given by solving the following Euler-Lagrange equation

$$\begin{aligned} &\left(GL \cdot \begin{bmatrix} u|_{P\setminus S} \\ 0 \end{bmatrix} + \gamma \cdot DW \cdot \begin{bmatrix} u|_{P\setminus S} \\ 0 \end{bmatrix} + \lambda GL^T \cdot D \cdot GL \cdot \begin{bmatrix} u|_{P\setminus S} \\ 0 \end{bmatrix} \right)(\boldsymbol{x}) \\ &\quad = \textstyle\sum_{\boldsymbol{y}\in S} w(\boldsymbol{x},\boldsymbol{y}) g(\boldsymbol{y}) + \gamma \sum_{\boldsymbol{y}\in S} w(\boldsymbol{y},\boldsymbol{x}) g(\boldsymbol{y}) - \lambda \left(GL^T \cdot D \cdot GL \begin{bmatrix} 0 \\ g \end{bmatrix} \right)(\boldsymbol{x}), \quad \boldsymbol{x} \in P\setminus S, \end{aligned} \tag{2.4}$$

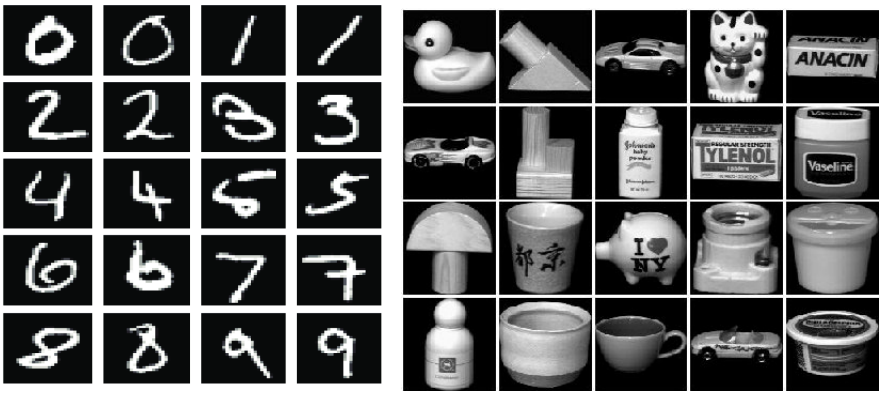


Figure 3: Some images in MNIST and COIL20 dataset.

where $DW = \mathrm{diag}(w_1, w_2, \ldots, w_{|P|})$ with $w_i = \sum_{\boldsymbol{y}\in S} w(\boldsymbol{x}_i, \boldsymbol{y})$ and $\gamma = \frac{|P|}{|S|}$ is the weighted coefficient in WNLL. The linear system above is symmetric positive definite and sparse which can be solved efficiently by iterative solvers such as the conjugate gradient method. We remark that, for (non-weighted) CURE method, we only need to replace matrix D above by identity matrix $Id_{|P|\times|P|}$. We summarize (We)CURE algorithm for missing data recovery in Algorithm 2.1.

Algorithm 2.1 (We)CURE for Missing Data Recovery

Require: Given point set $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\} \subset \mathbb{R}^d$ and a partially labeled set $S \subset P$, and given the function values of u on S, *i.e.* $u(\boldsymbol{x}) = g(\boldsymbol{x})$ for $\boldsymbol{x} \in S$.

Ensure: A recovered function u on P.

Calculate the weight matrix $W = (w(\boldsymbol{p}_i, \boldsymbol{p}_j))_{n\times n}$ and the graph Laplacian GL.

Set $DW = \mathrm{diag}([\sum_{j=1}^{m} w(\boldsymbol{p}_i, \boldsymbol{p}_j)]_{i=m+1:n})$.

Solving the linear system (2.4) for $u|_{P\setminus S}$.

3. CURE for Semi-Supervised Learning. Semi-supervised learning is a challenging and yet frequently encountered machine learning task. It can be formulated as a missing data recovery problem [53]. Given a data set $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\} \subset \mathbb{R}^d$, we assume there are totally l different classes. Let $S \subset P$ be a subset of P with labels, i.e

$$S = \bigcup_{i=1}^{l} S_i,$$

where $S_i \subset P$ is the subset with label i. It is typical for semi-supervised learning that $|S|$ is far less than $|P|$. The objective of semi-supervised learning is to extend labels to the entire data set P. Our algorithm is summarized in Algorithm 3.1.

We test WNLL, Weighted Nonlocal Total Variation (WNTV) [30], CURE, WeCURE on the MNIST dataset [29] of handwritten digits classification, COIL20 dataset [35] of object classification and ISOLET dataset [21] of spoken letter recognition. Some sample images from

Algorithm 3.1 (We)CURE for Semi-Supervised Learning

Require: Point set $P = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_n\} \subset \mathbb{R}^d$ and a partially labeled set $S = \bigcup_{i=1}^{l} S_i$.
Ensure: A complete label assignment $L : P \to \{1, 2, \ldots, l\}$
for $i = 1 : l$ **do**
Compute ϕ_i on P with the known observation

$$\phi_i(\boldsymbol{x}) = 1, \boldsymbol{x} \in S_i, \qquad \phi_i(\boldsymbol{x}) = 0, \boldsymbol{x} \in S \backslash S_i, \tag{3.1}$$

by Algorithm 2.1.
end for
for $\boldsymbol{x} \in P \backslash S$ **do**
Label $\boldsymbol{x}$ as following

$$L(\boldsymbol{x}) = k, \qquad \text{where} \quad k = \arg\max_{1 \le i \le l} \phi_i(\boldsymbol{x}) \tag{3.2}$$

end for

MNIST and COIL20 are shown in Figure 3. The MNIST dataset contains 70,000 gray-scale images of size 28 × 28 with 10 classes of digits going from 0 to 9. Each class contains 7,000 images. Each image can be seen as a point in a 784-dimension Euclidean space. The COIL20 dataset contains 20 objects, and each object has 72 images. The size of each image is 32 × 32 pixels, with 256 gray levels per pixel. Thus, each image is represented by a 1024-dimensional vector. The ISOLET dataset contains 150 subjects who spoke the name of each letter of the alphabet twice. The speakers are grouped into five sets of 30 speakers each and are referred to as isolet1 through isolet5. In our experiment, we use isolet1 which consists of 1560 samples with each sample represented by a 617-dimensional vector.

The weight function $w(\boldsymbol{x}, \boldsymbol{y})$ is constructed as

$$w(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{\sigma(\boldsymbol{x})^2}\right), \tag{3.3}$$

where $\sigma(\boldsymbol{x})$ is chosen to be the distance between $\boldsymbol{x}$ and its kth nearest neighbor ($k = 20$ in MNIST, $k = 15$ in COIL20 and ISOLET). To make the weight matrix sparse, the weight $w(\boldsymbol{x}, \boldsymbol{y})$ is truncated to the 50 nearest neighbors.

In our test on MNIST, we choose five different sampling rates to form the training set: labeling 700, 100, 70, 50 and 35 images in the whole dataset at random. For each sampling rate, we repeat the test results 10 times. In our test on COIL20 and ISOLET, we choose three different sampling rates to form the training set: labeling 2%, 5%, 10% at random. For each sampling rate, we repeat the test 10 times. Figure 4 shows the success rate of WNLL, CURE, and WeCURE method on MNIST dataset. The first five images of Figure 4 show the success rate for each sampling rate, while the last image shows the average success rate for each of the five sampling rate. It can be clearly observed that the proposed CURE and WeCURE outperform WNLL for all the tested cases. With a high sampling rate, WeCURE is comparable

with CURE, whereas WeCURE outperforms CURE in the cases with lower sampling rates. In terms of average success rate, both CURE and WeCURE outperform WNLL. We also compare (We)CURE with WNLL and Weighted Nonlocal Total Variation (WNTV) [30] in Table 1. It can be seen that (We)CURE significantly outperforms both WNLL and WNTV in cases with lower sample rates (50/70000, 100/70000). Table 2 shows the result on COIL20 and ISOLET dataset. It can be seen that WeCURE outperforms CURE and WNLL by 3% $\sim$ 4%.

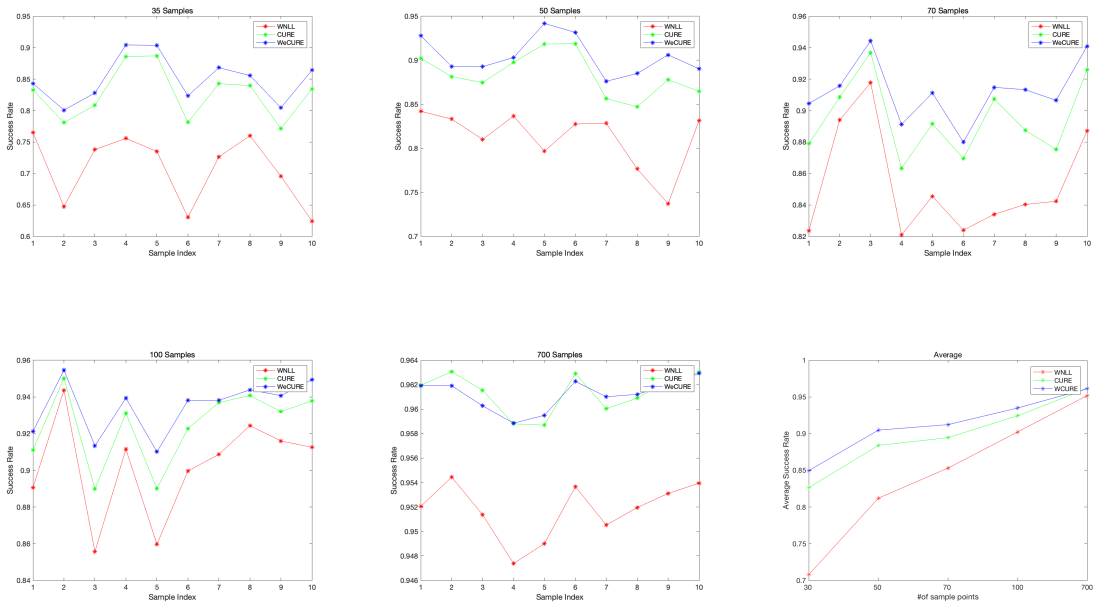


Figure 4: Comparisons of success rates by WNLL, CURE and WeCURE on MNIST.

Method	50/70000	100/70000	700/70000
WNLL [44]	73.60	87.84	93.25
WNTV [30]	78.35	89.86	94.08
CURE	88.40	92.42	96.13
WeCURE	90.48	93.49	96.12

Table 1: Classification accuracy in percentage for MNIST. The best results are in red and the second best results are in blue.

4. CURE for Image Inpainting. In this section, we apply (We)CURE to reconstruct the images with partially observed pixels. We adopt the assumption that image patches lie on a low dimensional and smooth manifold. Given an image $f \in \mathbb{R}^{m\times n}$, for any $(i,j) \in \{1,2,\ldots,m\}\times\{1,2,\ldots,n\}$, we define an $s_1 \times s_2$ image patch as

$$p_{ij}(f) = \{f(\tilde{i},\tilde{j}) : i-(s_1-1)/2 \le \tilde{i} \le i+(s_1-1)/2,\ j-(s_2-1)/2 \le \tilde{j} \le j+(s_2-1)/2\},$$

Method	COIL20			ISOLET		
	2%	5%	10%	2%	5%	10%
WNLL [44]	59.59	74.13	80.65	49.12	61.90	73.05
CURE	59.73	74.77	80.91	49.14	61.94	73.23
WeCURE	63.29	77.65	84.76	52.65	64.92	76.50

Table 2: Classification accuracy in percentage for COIL20 and ISOLET. The best results are in red and the second best results are in blue.

where we assume s_1 and s_2 are odd integers and we adopt reflective boundary conditions for (i, j) near image boundary. Define the patch set $P(f)$ as the collection of all patches:

$$P(f) = \{p_{ij}(f) : (i, j) \in \{1, 2, \dots, m\} \times \{1, 2, \times, n\}\} \subset \mathbb{R}^d, \quad d = s_1 \cdot s_2.$$

Define a function u on $P(f)$ as

$$u(p_{ij}(f)) = f(i, j),$$

where $f(i, j)$ is the intensity of image f at pixel (i, j).

Now, suppose we only observe the image on a subset of pixels $\Omega \subset \{(i, j) : 1 \le i \le m, 1 \le j \le n\}$. We would like to recover the entire image f from the observed data $f|_\Omega$. This problem can be recast as the interpolation of the function u on the patch set $P(f)$ with u being given in $S \subset P(f)$, $S = \{p_{ij}(f) : (i, j) \in \Omega\}$. This falls into the general algorithmic framework of (We)CURE for missing data recovery (Algorithm 2.1). Notice that the patch set $P(f)$ is unknown. Thus, we need to iterative update the patch set $P(f)$. We summarize the (We)CURE algorithm for this problem in Algorithm 4.1.

Algorithm 4.1 Subsampled image restoration By WeCURE

Require: A subsampled image $f|_\Omega$
Ensure: A recovered image u
Generate initial image u^0
while not converge **do**
1: Generate the semi-local patch set $\bar{P}(u^n)$ from current image u^n and get corresponding labeled set $S^n \subset \bar{P}(u^n)$
2: Update the image by computing u^{n+1} on $P(u^n)$, with the known observation

$$u^{n+1}(\boldsymbol{x}) = f(\boldsymbol{x}), \quad \boldsymbol{x} \in S^n,$$

by Algorithm 2.1.
3: $n \leftarrow n + 1$.
end while
$u = u^n$

The weight function $w(\boldsymbol{x}, \boldsymbol{y})$ is chosen as (3.3). Here, $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{d+2}$ are vectorized semi-local patches and $\sigma(\boldsymbol{x})$ is chosen to be the distance between $\boldsymbol{x}$ and its 20th nearest neighbor. To make the weight matrix sparse, the weight is truncated to the 50 nearest neighbors. In the

Figure 5: Set12: 12 widely used testing images.

semi-local patches, the local coordinate is normalized to have the same amplitude as the image intensity,

$$(\bar{P}\boldsymbol{u})(\boldsymbol{x}) = [(P\boldsymbol{u})(\boldsymbol{x}), \lambda\bar{\boldsymbol{x}}]$$

with

$$\bar{\boldsymbol{x}} = \left(\frac{x_1\|(f|_\Omega)\|_\infty}{m}, \frac{x_2\|(f|_\Omega)\|_\infty}{n}\right),$$

where $\boldsymbol{x} = (x_1, x_2)$ and m, n are the size of the image. The purpose of introducing semi-local patches is to constrain the search space to a local area. The larger λ leads to smaller search space making the searching quicker, while smaller λ leads to global search and make more accurate results. Thus following [44] we gradually reduce λ by $\lambda^{k+1} = \max(\lambda^k - 1, 3)$ and initialization $\lambda = 10$.

We apply our algorithm to 12 widely used test images (Figure 5). In our experiment, we select the patch size to be 11×11. For each patch, the nearest neighbors are obtained by using an approximate nearest neighbor (ANN) search algorithm. We use a k-d tree approach as well as an ANN search algorithm to reduce the computational cost. The linear system in weighted nonlocal Laplacian and graph Laplacian is solved by the conjugate gradient method. We use the solution of WNLL after 6 steps as the initialization of our algorithm to get a proper initial guess of the similarity relationships between different groups. The initial image of WNLL is obtained by filling the missing pixels with random numbers which satisfy a Gaussian distribution, where μ_0 is the mean of $f|_\Omega$ and σ_0 is the standard deviation of $f|_\Omega$. The number of iterations of LDMM, WNLL, CURE, WeCURE are 10, 10, 8, 8, respectively.

Quality of the restored images is measured by PSNR and SSIM. PSNR is defined as

$$\mathrm{PSNR}(f, f^*) = -20\,\log_{10}(\|f - f^*\|/255) \tag{4.1}$$

where f^* is the ground truth. SSIM is defined as a multiplication of three terms that quantifies similarity of luminance, contrast and structure. It takes the following form

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma, \tag{4.2}$$

where

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1},\ c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2},\ s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}, \tag{4.3}$$

where $\mu_x, \mu_y, \sigma_x, \sigma_x$ and σ_{xy} are the local means, standard deviations and cross-covariance for image x, y.

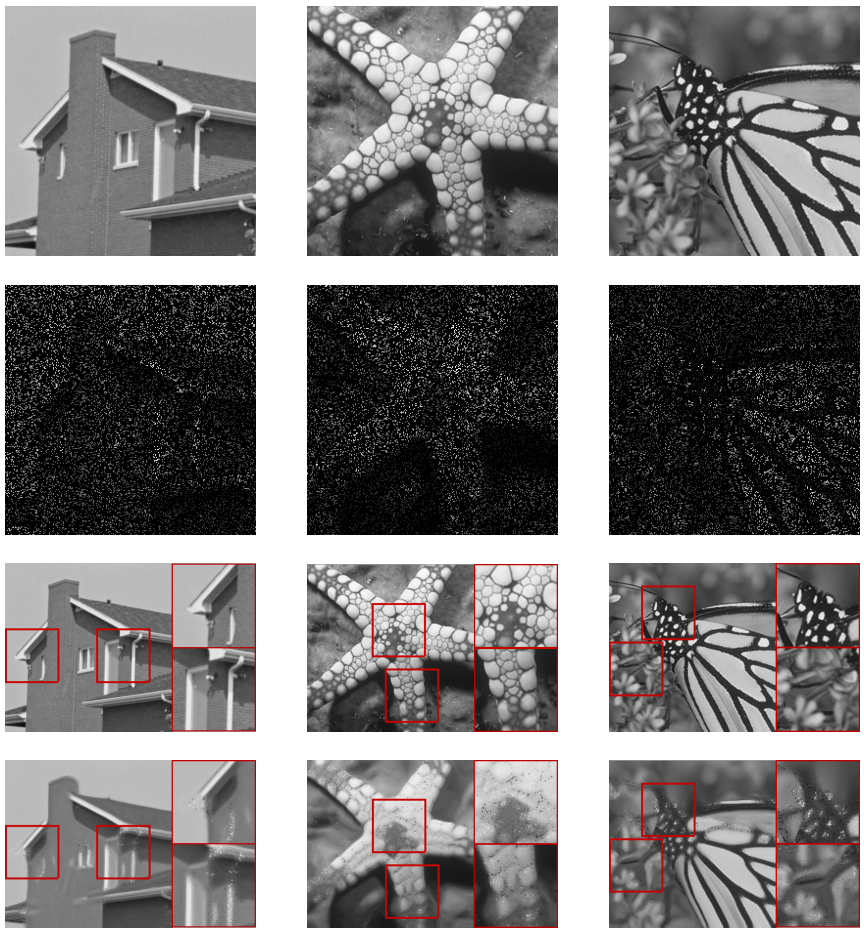
The numerical results are shown in Table 3 and Table 4. For qualitative comparisons, Figure 6 shows the inpainting results of 3 images from Set12 dataset at 15% sample rate. Figure 7 shows the inpainting results at 20% sample rate. As we can see, WeCURE gives much better results than WNLL both visually and in terms of PSNR and SSIM. We observe that (We)CURE can well recover texture and preserve sharp image features such as edges, though it also introduces mild artifacts in smooth regions. This is why (We)CURE significantly outperforms WNLL in terms of SSIM.

Images	C.man	House	Peppers	Starfish	Monarch	Airplane	Parrot	Lena	Barbara	Boat	Man	Couple	Average
Sample Rate							10%						
LDMM	19.9329	24.8723	20.6103	19.9285	19.3395	19.9612	19.5449	26.1005	23.3176	22.6681	23.9415	22.7225	21.9117
WNLL	**21.9993**	28.3325	23.3210	22.2705	22.4218	**21.7954**	21.6121	28.5089	26.3732	24.8116	**25.8126**	25.0263	24.3571
CURE	21.7095	28.3023	23.3315	22.0185	22.0650	21.4078	21.5080	28.3013	26.3031	24.6798	25.7207	24.9033	24.1876
WeCURE	21.8571	**28.7967**	**23.7416**	**22.3540**	**22.5829**	21.4335	**21.7753**	**28.7926**	**26.7155**	**25.0060**	25.7145	**25.1940**	**24.4970**
Sample Rate							15%						
LDMM	21.0948	26.4075	21.6434	20.9887	20.9843	21.0712	21.3412	27.7591	25.6175	23.8791	25.1269	24.0065	23.3267
WNLL	**23.3052**	29.1647	25.0635	23.5147	23.7171	22.7292	22.5851	29.5856	27.7837	25.8633	26.9433	26.2245	25.5400
CURE	22.8514	29.5745	25.1007	23.4509	23.8326	22.5211	22.4579	29.6253	27.7315	25.7653	26.9278	26.1798	25.5016
WeCURE	23.0993	**30.9540**	**25.7840**	**24.0722**	**24.2587**	**22.8246**	**22.8708**	**30.1331**	**28.5615**	**26.2943**	**27.3484**	**26.7266**	**26.0773**
Sample Rate							20%						
LDMM	21.9057	28.2924	22.7767	22.6264	22.4175	22.1073	21.9409	28.9160	26.8121	24.8777	26.2350	25.0044	24.4927
WNLL	23.9478	30.8222	25.8068	24.5382	24.6738	23.8359	23.2844	30.5140	28.7357	26.6614	27.7806	26.7532	26.4462
CURE	23.7846	31.4606	25.7513	24.7232	24.8360	23.7147	23.5282	30.6271	28.9715	26.6736	27.8198	26.8165	26.5589
WeCURE	**24.5007**	**32.1789**	**26.6428**	**25.3982**	**25.5151**	**24.1406**	**24.0625**	**31.3711**	**29.7794**	**27.3033**	**28.3473**	**27.4934**	**27.2278**

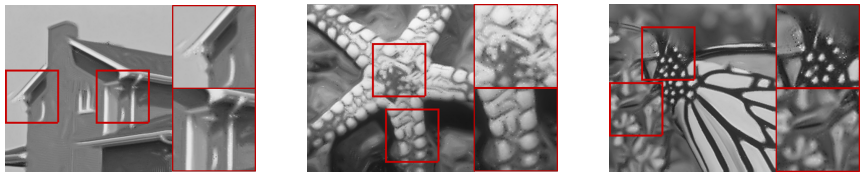
Table 3: The PSNR(dB) results of different methods on Set12 dataset with sampling rate 10%, 15% and 20%. The best results are indicated in red and are highlighted in bold. The second best results are indicated in blue and are highlighted by underline.

Images	C.man	House	Peppers	Starfish	Monarch	Airplane	Parrot	Lena	Barbara	Boat	Man	Couple	Average
Sample Rate							10%						
LDMM	0.2677	0.3406	0.4406	0.3856	0.4870	0.3338	0.4560	0.4508	0.4881	0.3121	0.3469	0.3389	0.3874
WNLL	0.3557	0.4236	0.5681	0.5415	0.6523	0.4352	0.5680	0.5316	0.6308	0.4383	0.4787	0.5123	0.5113
CURE	0.3591	0.4337	0.5849	0.5382	0.6537	0.4324	0.5733	0.5356	0.6392	0.4409	0.4817	0.5240	0.5164
WeCURE	**0.3726**	**0.4397**	**0.6042**	**0.5721**	**0.6842**	**0.4448**	**0.5953**	**0.5402**	**0.6572**	**0.4628**	**0.5051**	**0.5476**	**0.5355**
Sample Rate							15%						
LDMM	0.3622	0.4288	0.5308	0.4848	0.5986	0.4252	0.5464	0.5382	0.6164	0.4187	0.4483	0.4619	0.4884
WNLL	0.4456	0.5053	0.6380	0.6196	0.7076	0.5052	0.6247	0.5931	0.6964	0.5130	0.5544	0.5911	0.5828
CURE	0.4464	0.5294	0.6610	0.6294	0.7299	0.5115	0.6435	0.5994	0.7068	0.5226	0.5637	0.6067	0.5959
WeCURE	**0.4577**	**0.5459**	**0.6766**	**0.6658**	**0.7473**	**0.5273**	**0.6621**	**0.6102**	**0.7275**	**0.5462**	**0.5939**	**0.6308**	**0.6159**
Sample Rate							20%						
LDMM	0.4385	0.5148	0.5980	0.5783	0.6692	0.5003	0.6074	0.5997	0.6840	0.5003	0.5295	0.5501	0.5642
WNLL	0.4970	0.5735	0.6856	0.6691	0.7439	0.5684	0.6673	0.6376	0.7373	0.5722	0.6062	0.6364	0.6329
CURE	0.5063	0.6044	0.7051	0.6889	0.7687	0.5847	0.6850	0.6457	0.7515	0.5882	0.6203	0.6571	0.6505
WeCURE	**0.5270**	**0.6167**	**0.7241**	**0.7214**	**0.7859**	**0.6009**	**0.7017**	**0.6570**	**0.7683**	**0.6093**	**0.6492**	**0.6806**	**0.6702**

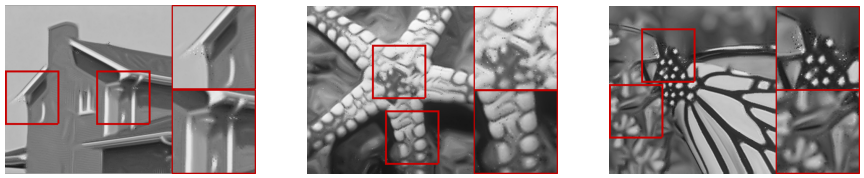
Table 4: The SSIM results of different methods on Set12 dataset with sampling rate 10%, 15% and 20%. The best results are indicated in red and are highlighted in bold. The second best results are indicated in blue and are highlighted by underline.



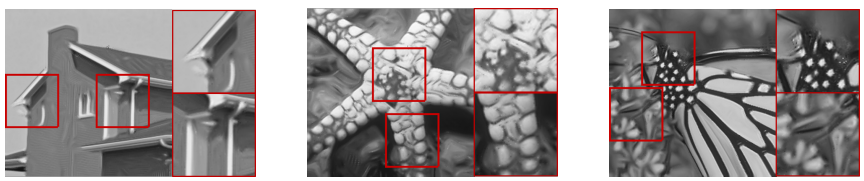
PSNR=26.41dB,SSIM=0.42 PSNR=20.99dB,SSIM=0.48 PSNR=20.98dB,SSIM=0.60



PSNR=29.16dB,SSIM=0.50 PSNR=23.51dB,SSIM=0.62 PSNR=23.71dB,SSIM=0.71

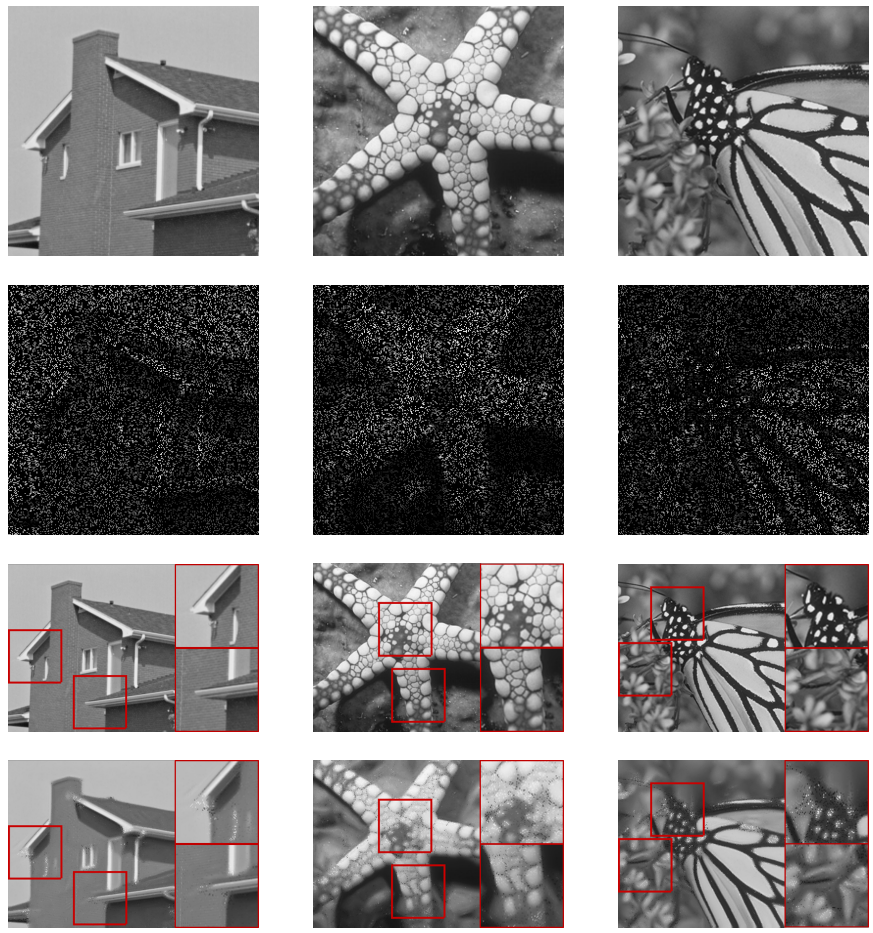


PSNR=29.57dB,SSIM=0.53 PSNR=23.45dB,SSIM=0.63 PSNR=23.83dB,SSIM=0.73



PSNR=30.95dB,SSIM=0.55 PSNR=24.07dB,SSIM=0.68 PSNR=24.26dB,SSIM=0.75

Figure 6: From top to bottom: original image, 15% subsample, ground-truth, LDMM, WNLL, CURE, WeCURE



PSNR=28.29dB,SSIM=0.51 PSNR=22.63dB,SSIM=0.58 PSNR=22.42dB,SSIM=0.67

PSNR=30.82dB,SSIM=0.57 PSNR=24.54dB,SSIM=0.67 PSNR=24.67dB,SSIM=0.74

PSNR=31.46dB,SSIM=0.60 PSNR=24.72dB,SSIM=0.68 PSNR=24.83dB,SSIM=0.77

PSNR=32.18dB,SSIM=0.62 PSNR=25.39dB,SSIM=0.72 PSNR=25.51dB,SSIM=0.79

Figure 7: From top to bottom: original image, 20% subsample, ground-truth, LDMM, WNLL, CURE, WeCURE

As for the computational time, we include the time of initialization and thus WeCURE (using 6 steps of WNLL as initialization) is approximately 1.5 times slower than WNLL to inpaint an 256×256 image at 15% sample rate, as shown in Table 5. The computation of the weight matrix is very expensive in all the methods we test and take more than 90% of the total time. All experiments are implemented on MATLAB R2018b running on a laptop equipped with CPU Intel i5-7200U 2.5 GHz.

	LDMM	WNLL	CURE	WeCURE
CPU Time/s	102.35	101.62	153.77	154.23

Table 5: The CPU time of different methods on 256×256 images in Set12 dataset with sampling rate 15%. The best result is indicated in red. The second best result is indicated in blue.

5. Asymptotic Analysis. In this section, we provide an asymptotic analysis of the proposed numerical scheme for WeCURE model using Γ-convergence. The idea of the proof is sketched as follows. We first fix the bandwidth of the kernel and consider our scheme as an integral scheme of a non-local functional. Then, we reduce the bandwidth of the kernel to zero to show that the non-local functional approximates the original WeCURE functional. The general idea and the notation we use for the proof mostly follows [45, 48]. Since the proofs of the theorems that will proposed later can be carried out similarly as in [45, 48], for better readability, we shall omit the proofs and only present the theorems. We also note that, in a recent paper [23], the authors also established a Γ-convergence result of the Biharmonic equation. The difference between their work and ours is mainly the problem setting. In their paper, labeled data is treated as the boundary condition, while in our paper, we consider the labeled data as samples from the data distribution and the ratio of the number of the labeled and unlabeled data is a fixed factor. Under this setting, we will show that weights of WeCURE are crucial to achieving convergence.

Let $P = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$ with $\boldsymbol{x}_i$ being uniformly sampled from Ω, where Ω is an open bounded domain in $\mathbb{R}^d$. Let $S = \{\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \cdots, \boldsymbol{x}_{i_m}\}$ the set of labeled points where $\boldsymbol{x}_{i_j}$ is uniformly sampled from P. We assume that the ratio $\gamma = \frac{n}{m}$ is fixed. Let $b : \Omega \to \mathbb{R}$ be a function whose value is only known at labeled points. The empirical measure of data points is given by $\mu_n = \frac{1}{n}\sum_{i=1}^n \delta_{\boldsymbol{x}_i}$. We consider a graph with vertices $V = P$ and edge weights $W_{ij} = \eta_\varepsilon(\boldsymbol{x}_i - \boldsymbol{x}_j)$ where $\eta_\varepsilon(\boldsymbol{x}) := \eta_\varepsilon(|\boldsymbol{x}|) = \frac{1}{\varepsilon^d}\eta\left(\frac{|\boldsymbol{x}|}{\varepsilon}\right), \eta : [0, +\infty) \to [0, +\infty)$ is a radially symmetric function which satisfies the following assumptions:
(A1) $\eta(0) > 0$ and η is continuous at 0.
(A2) η is non-increasing.
(A3) η has compact support. If $|r| > \alpha$ then $\eta(r) = 0$.

The discrete WeCURE model is given by (the weight is $\gamma-1$, not γ in previous sections)

(5.1a)
$$\mathrm{WeCURE}_{n,\varepsilon}(u)=\frac{1}{\varepsilon^2}\frac{1}{(n-m)^2}\sum_{i\in P\setminus S}\sum_{j\in P\setminus S}W_{ij}\left(u(\boldsymbol{x}_i)-u(\boldsymbol{x}_j)\right)^2$$

(5.1b)
$$+\frac{2}{(n-m)m}\sum_{i\in S}\sum_{j\in P\setminus S}W_{ij}\left(u(\boldsymbol{x}_i)-b(\boldsymbol{x}_j)\right)^2$$

(5.1c)
$$+\frac{\lambda}{\varepsilon^4}\left(\frac{1}{(n-m)^3}\sum_{i\in P\setminus S}\left(\sum_{j\in P\setminus S}W_{ij}\left(u(\boldsymbol{x}_i)-u(\boldsymbol{x}_j)\right)\right)^2\right.$$

(5.1d)
$$+\frac{2\varepsilon^2}{(n-m)^2m(\gamma-1)}\sum_{i\in P\setminus S}\left(\sum_{j\in P\setminus S}W_{ij}\left(u(\boldsymbol{x}_i)-u(\boldsymbol{x}_j)\right)\sum_{j\in S}W_{i,j}\left(u(\boldsymbol{x}_i)-b(\boldsymbol{x}_j)\right)\right)$$

(5.1e)
$$+\frac{\varepsilon^4}{(n-m)m^2(\gamma-1)^2}\sum_{i\in P\setminus S}\left(\sum_{j\in S}W_{ij}\left(u(\boldsymbol{x}_i)-b(\boldsymbol{x}_j)\right)\right)^2$$

(5.1f)
$$+\frac{\varepsilon^4}{(n-m)^2m}\sum_{i\in S}\left(\sum_{j\in P\setminus S}W_{ij}\left(b(\boldsymbol{x}_i)-u(\boldsymbol{x}_j)\right)\right)^2$$

(5.1g)
$$\left.+\frac{2\varepsilon^2}{(n-m)m^2(\gamma-1)}\sum_{i\in S}\left(\sum_{j\in P\setminus S}W_{ij}\left(b(\boldsymbol{x}_i)-u(\boldsymbol{x}_j)\right)\sum_{j\in S}W_{ij}\left(b(\boldsymbol{x}_i)-b(\boldsymbol{x}_j)\right)\right)\right)$$

The nonlocal continuum WeCURE model is given by

(5.2a)
$$\mathrm{WeCURE}_{\varepsilon}(u)=\frac{1}{\varepsilon^2}\int_\Omega\int_\Omega\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\left(u(\boldsymbol{x})-u(\boldsymbol{y})\right)^2d\boldsymbol{y}d\boldsymbol{x}$$

(5.2b)
$$+2\int_\Omega\int_\Omega\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\left(u(\boldsymbol{x})-b(\boldsymbol{y})\right)^2d\boldsymbol{y}d\boldsymbol{x}$$

(5.2c)
$$+\frac{\lambda}{\varepsilon^4}\left(\int_\Omega\left(\int_\Omega\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\left(u(\boldsymbol{x})-u(\boldsymbol{y})\right)d\boldsymbol{y}\right)^2d\boldsymbol{x}\right.$$

(5.2d)
$$+\frac{2\varepsilon^2}{\gamma-1}\int_\Omega\left(\int_\Omega\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\left(u(\boldsymbol{x})-u(\boldsymbol{y})\right)d\boldsymbol{y}\int_\Omega\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\left(u(\boldsymbol{x})-b(\boldsymbol{y})\right)d\boldsymbol{y}\right)d\boldsymbol{x}$$

(5.2e)
$$+\frac{\varepsilon^4}{(\gamma-1)^2}\int_\Omega\left(\int_\Omega\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\left(u(\boldsymbol{x})-b(\boldsymbol{y})\right)d\boldsymbol{y}\right)^2d\boldsymbol{x}$$

(5.2f)
$$+\varepsilon^4\int_\Omega\left(\int_\Omega\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\left(b(\boldsymbol{x})-u(\boldsymbol{y})\right)d\boldsymbol{y}\right)^2d\boldsymbol{x}$$

(5.2g)
$$\left.+\frac{2\varepsilon^2}{\gamma-1}\int_\Omega\left(\int_\Omega\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\left(b(\boldsymbol{x})-u(\boldsymbol{y})\right)d\boldsymbol{y}\int_\Omega\eta_\varepsilon(\boldsymbol{x}-\boldsymbol{y})\left(b(\boldsymbol{x})-b(\boldsymbol{y})\right)d\boldsymbol{y}\right)d\boldsymbol{x}\right)$$

The (local) continuum WeCURE model is given by

$$\text{WeCURE}(u) = 2\sigma_\eta \int_\Omega |\nabla u(\boldsymbol{x})|^2 d\boldsymbol{x} \tag{5.3a}$$

$$+ 2\int_\Omega (u(\boldsymbol{x}) - b(\boldsymbol{x}))^2 d\boldsymbol{x} \tag{5.3b}$$

$$+ \lambda\left(\sigma_\eta^2 \int_\Omega (\Delta u(\boldsymbol{x}))^2 d\boldsymbol{x}\right. \tag{5.3c}$$

$$+ \frac{2\sigma_\eta}{\gamma-1}\int_\Omega (-\Delta u(\boldsymbol{x}))(u(\boldsymbol{x}) - b(\boldsymbol{x}))\, d\boldsymbol{x} \tag{5.3d}$$

$$+ \left(\frac{1}{(\gamma-1)^2} + 1\right)\int_\Omega (u(\boldsymbol{x}) - b(\boldsymbol{x}))^2 d\boldsymbol{x} \tag{5.3e}$$

$$\left. + \frac{2\sigma_\eta}{\gamma-1}\int_\Omega (b(\boldsymbol{x}) - u(\boldsymbol{x}))(-\Delta b(\boldsymbol{x}))\, d\boldsymbol{x}\right) \tag{5.3f}$$

$$= \lambda\sigma_\eta^2 \int_\Omega (\Delta u(\boldsymbol{x}))^2 d\boldsymbol{x} \tag{5.3g}$$

$$+ 2\sigma_\eta \int_\Omega |\nabla u(\boldsymbol{x})|^2 d\boldsymbol{x} \tag{5.3h}$$

$$+ \frac{2\lambda\sigma_\eta}{\gamma-1}\int_\Omega |\nabla(u(\boldsymbol{x}) - b(\boldsymbol{x}))|^2 d\boldsymbol{x} \tag{5.3i}$$

$$+ \left(2 + \lambda + \frac{\lambda}{(\gamma-1)^2}\right)\int_\Omega (u(\boldsymbol{x}) - b(\boldsymbol{x}))^2 d\boldsymbol{x} \tag{5.3j}$$

where $\sigma_\eta = \frac{1}{2}\int_{\mathbb{R}^d} \eta(\boldsymbol{h})|h_1|^2 d\boldsymbol{h}$, h_1 is the first coordinate of vector $\boldsymbol{h}$.

Remark 5.1. The models introduced above contain the corresponding CURE models as special cases. To see this, we can simply modify some coefficients in the WeCURE models and replace the term $\int_\Omega |\nabla(u(\boldsymbol{x}) - b(\boldsymbol{x}))|^2 d\boldsymbol{x}$ by $\int_\Omega |\nabla(u(\boldsymbol{x}) - c\cdot b(\boldsymbol{x}))|^2 d\boldsymbol{x}$ ($c \neq 1$ being a certain constant).

We are now ready to present the main theorems of this section.

Theorem 5.2. *Let $\Omega \subset \mathbb{R}^d$, $d \geq 2$ be an open, bounded, connected set with Lipschitz boundary. Let $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \cdots$ be a sequence of i.i.d random points uniformly sampled from Ω. $S = \{\boldsymbol{x}_{i_1}, \boldsymbol{x}_{i_2}, \cdots, \boldsymbol{x}_{i_m} : \boldsymbol{x}_{i_j}$ uniformly sampled from $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n, \cdots\}$ is the set of labeled points whose value is given by $b(\boldsymbol{x}_{i_j})$. Assume the kernel η satisfies conditions (A1)-(A3). Then* $\text{WeCURE}_{n,\varepsilon}$ *Γ-converge to* $\text{WeCURE}_\varepsilon$ *as $n \to \infty$ in the TL^2 sense.*

Theorem 5.3. *Under the assumptions of Theorem 5.2,* $\text{WeCURE}_\varepsilon$ *Γ-converge to* $\text{WeCURE}(u)$ $\left(u \in H_0^2(\Omega)\right)$ *as $\varepsilon \to 0$ in $L^2(\Omega)$ metric.*

Theorem 5.4. *(Compactness)Under the assumptions of Theorem 5.2,* $\{\text{WeCURE}_\varepsilon\}_{\varepsilon>0}$ *satisfy the compactness property with respect to the $L^2(\Omega)$ metric.*

6. Conclusion and Future Work. In this paper, we proposed to use both low dimensionality and smoothness of the underlying data manifold as a regularizer for missing data

recovery. For that, we introduced curvature regularization (CURE) and a weighted version of it (WeCURE). Compared to related models such as LDMM, WNLL, and WNTV, the new regularization is proven to be more effective for semi-supervised learning and image inpainting on some datasets.

There are plenty of future directions worth exploring. For modelling, a natural question is whether different curvatures can also serve as good smoothing regularizers regularizer for data manifolds and how are they different from the one we chose for CURE? Can these curvatures be easily computed? How does CURE work for other tasks of missing data recovery? Furthermore, convergence analysis of solving the Biharmonic equation (2.4) on manifold also needs to be studied. Note that [10] proposed a directional operator splitting scheme to solve the biharmonic equation on regular grids, while our algorithm solves it on point clouds. Due to a lack of understanding of the numerical methods for the biharmonic equation on point clouds, it prohibited us from generalizing CURE to generic inverse problems.

Acknowledgments. Bin Dong is supported in part by NSFC 11671022 and Beijing Natural Science Foundation (Z180001). Haocheng Ju is supported by the Elite Undergraduate Training Program of the School of Mathematical Sciences at Peking University. Zuoqiang Shi is supported by NSFC 11671005. We would also like to thank Dr. Wei Zhu for his valuable comments and kindly sharing the codes of both LDMM and LDMM+WNLL for comparisons.

REFERENCES

[1] S. Agarwal, K. Branson, and S. Belongie, *Higher order learning with graphs*, in Proceedings of the 23rd international conference on Machine learning, ACM, 2006, pp. 17–24.
[2] C. Bao, B. Dong, L. Hou, Z. Shen, X. Zhang, and X. Zhang, *Image restoration by minimizing zero norm of wavelet frame coefficients*, Inverse problems, 32 (2016), p. 115004.
[3] A. L. Bertozzi and A. Flenner, *Diffuse interface models on graphs for classification of high dimensional data*, Multiscale Modeling & Simulation, 10 (2012), pp. 1090–1118.
[4] K. Bredies, K. Kunisch, and T. Pock, *Total generalized variation*, SIAM Journal on Imaging Sciences, 3 (2010), pp. 492–526.
[5] A. Buades, B. Coll, and J.-M. Morel, *A non-local algorithm for image denoising*, in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 2, IEEE, 2005, pp. 60–65.
[6] A. Buades, B. Coll, and J.-M. Morel, *A review of image denoising algorithms, with a new one. multiscale model*, Simul, 4 (2005), p. 490530.
[7] A. Buades, B. Coll, and J.-M. Morel, *Neighborhood filters and PDEs*, Numer. Math, 105 (2006), p. 134.
[8] J.-F. Cai, R. H. Chan, and Z. Shen, *Simultaneous cartoon and texture inpainting*, Inverse Probl. Imaging, 4 (2010), pp. 379–395.
[9] J.-F. Cai, S. Osher, and Z. Shen, *Split Bregman methods and frame based image restoration*, Multiscale modeling & simulation, 8 (2009), pp. 337–369.
[10] L. Calatroni, B. Düring, and C.-B. Schönlieb, *ADI SPLITTING SCHEMES FOR A FOURTH-ORDER NONLINEAR PARTIAL DIFFERENTIAL EQUATION FROM IMAGE PROCESSING*, DYNAMICAL SYSTEMS, 34 (2014).
[11] J. Calder and D. Slepcev, *Properly-weighted graph laplacian for semi-supervised learning*, arXiv preprint arXiv:1810.04351, (2018).

[12] R. H. Chan, T. F. Chan, L. Shen, and Z. Shen, *Wavelet algorithms for high-resolution image reconstruction*, SIAM Journal on Scientific Computing, 24 (2003), pp. 1408–1432.
[13] T. Chan, A. Marquina, and P. Mulet, *High-order total variation-based image restoration*, SIAM Journal on Scientific Computing, 22 (2000), pp. 503–516.
[14] F. Chung, *Spectral Graph Theory*, American Mathematical Society, (1997).
[15] R. R. Coifman and S. Lafon, *Diffusion maps*, Applied and computational harmonic analysis, 21 (2006), pp. 5–30.
[16] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, *Image denoising with block-matching and 3d filtering*, in Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning, vol. 6064, International Society for Optics and Photonics, 2006, p. 606414.
[17] A. Danielyan, V. Katkovnik, and K. Egiazarian, *BM3D frames and variational image deblurring*, IEEE Transactions on Image Processing, 21 (2012), pp. 1715–1728.
[18] I. Daubechies, *Ten lectures on wavelets*, vol. 61, Siam, 1992.
[19] B. Dong, *Sparse representation on graphs by tight wavelet frames and applications*, Applied and Computational Harmonic Analysis, 42 (2017), pp. 452–479.
[20] B. Dong and Z. Shen, *MRA-based wavelet frames and applications: Image segmentation and surface reconstruction*, in Independent Component Analyses, Compressive Sampling, Wavelets, Neural Net, Biosystems, and Nanoengineering X, vol. 8401, International Society for Optics and Photonics, 2012, p. 840102.
[21] D. Dua and C. Graff, *UCI machine learning repository*, 2017, http://archive.ics.uci.edu/ml.
[22] G. Easley, D. Labate, and W.-Q. Lim, *Sparse directional image representations using the discrete shearlet transform*, Applied and Computational Harmonic Analysis, 25 (2008), pp. 25–46.
[23] A. El Alaoui, X. Cheng, A. Ramdas, M. J. Wainwright, and M. I. Jordan, *Asymptotic behavior of ℓ_p-based laplacian regularization in semi-supervised learning*, in Conference on Learning Theory, 2016, pp. 879–906.
[24] V. Fedorov, G. Facciolo, and P. Arias, *Variational framework for non-local inpainting*, Image Processing On Line, 5 (2015), pp. 362–386.
[25] G. Gilboa and S. Osher, *Nonlocal linear image regularization and supervised segmentation*, Multiscale Model. Simul, 6 (2007), p. 595630.
[26] G. Gilboa and S. Osher, *Nonlocal operators with applications to image processing*, Multiscale Modeling & Simulation, 7 (2008), pp. 1005–1028.
[27] S. Gu, L. Zhang, W. Zuo, and X. Feng, *Weighted nuclear norm minimization with application to image denoising*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2862–2869.
[28] R. Lai and J. Li, *Manifold based low-rank regularization for image restoration and semi-supervised learning*, Journal of Scientific Computing, 74 (2018), pp. 1241–1263.
[29] Y. LeCun, *The MNIST database of handwritten digits*, http://yann. lecun. com/exdb/mnist/, (1998).
[30] H. Li, Z. Shi, and X.-P. Wang, *Weighted Nonlocal Total Variation in Image Processing*, arXiv preprint, arXiv:1801.10441, (2019).
[31] Z. Li and Z. Shi, *A convergent point integral method for isotropic elliptic equations on point cloud*, SIAM: Multiscale Modeling Simulation, 14 (2016), p. 874905.
[32] W.-Q. Lim, *The discrete shearlet transform: a new directional transform and compactly supported shearlet frames.*, IEEE Trans. Image Processing, 19 (2010), pp. 1166–1180.
[33] F. Manfio and F. Vitório, *Minimal immersions of Riemannian manifolds in products of space forms*, Journal of Mathematical Analysis and Applications, 424 (2015), pp. 260–268.
[34] B. Nadler, N. Srebro, and X. Zhou, *Semi-supervised learning with the graph laplacian: The limit of infinite unlabelled data*, Advances in neural information processing systems, 22 (2009), pp. 1330–1338.
[35] S. A. Nene, S. K. Nayar, and H. Murase, *Columbia object image library (coil-20*, tech. report, 1996.
[36] A. Newson, A. Almansa, Y. Gousseau, and P. Pérez, *Non-local patch-based image inpainting*, Image Processing On Line, 7 (2017), pp. 373–385.
[37] S. Osher, Z. Shi, and W. Zhu, *Low dimensional manifold model for image processing*, technical report, cam report 16-04, UCLA, 2016.
[38] K. Papafitsoros and C.-B. Schönlieb, *A combined first and second order variational approach for image reconstruction*, Journal of mathematical imaging and vision, 48 (2014), pp. 308–338.

[39] G. Peyré, *Manifold models for signals and images*, Computer Vision and Image Understanding, 113 (2009), pp. 249–260.
[40] S. Roth and M. J. Black, *Fields of experts: A framework for learning image priors*, in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 2, IEEE, 2005, pp. 860–867.
[41] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D: nonlinear phenomena, 60 (1992), pp. 259–268.
[42] Q. Shan, J. Jia, and A. Agarwala, *High-quality motion deblurring from a single image*, in Acm transactions on graphics (tog), vol. 27, ACM, 2008, p. 73.
[43] J. Shen, S. H. Kang, and T. F. Chan, *Euler's elastica and curvature-based inpainting*, SIAM journal on Applied Mathematics, 63 (2003), pp. 564–592.
[44] Z. Shi, S. Osher, and W. Zhu, *Weighted Nonlocal Laplacian on Interpolation From Sparse Data*, Journal of Scientific Computing, 73 (2017), pp. 1164–1177.
[45] D. Slepcev and M. Thorpe, *Analysis of p-laplacian regularization in semisupervised learning*, SIAM Journal on Mathematical Analysis, 51 (2019), pp. 2085–2120.
[46] J.-L. Starck, E. J. Candès, and D. L. Donoho, *The curvelet transform for image denoising*, IEEE Transactions on image processing, 11 (2002), pp. 670–684.
[47] M. Stephane, *A wavelet tour of signal processing*, The Sparse Way, (1999).
[48] N. G. Trillos and D. Slepčev, *Continuum limit of total variation on point clouds*, Archive for rational mechanics and analysis, 220 (2016), pp. 193–241.
[49] N. G. Trillos and D. Slepčev, *A variational approach to the consistency of spectral clustering*, Applied and Computational Harmonic Analysis, 45 (2018), pp. 239–281.
[50] Y. Zhang, B. Dong, and Z. Lu, ℓ_0 *minimization for wavelet frame based image restoration*, Mathematics of Computation, 82 (2013), pp. 995–1015.
[51] S. C. Zhu and D. Mumford, *Prior learning and Gibbs reaction-diffusion*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 19 (1997), pp. 1236–1250.
[52] W. Zhu, Q. Qiu, J. Huang, R. Calderbank, G. Sapiro, and I. Daubechies, *LDMNet: Low dimensional manifold regularized neural networks*, in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[53] X. Zhu, Z. Ghahramani, and J. Lafferty, *Semi-supervised Learning Using Gaussian Fields and Harmonic Functions*, in Machine Learning, Proceedings of the Twentieth International Conference ICML, Washington, DC, USA, 2003, p. 912919.