This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3136488, IEEE Open Journal of Signal Processing

Learning of Continuous and Piecewise-Linear Functions with Hessian Total-Variation Regularization

Joaquim Campos, Shayan Aziznejad, and Michael Unser, *Fellow, IEEE*

We develop a novel 2D functional learning framework that employs a sparsity-promoting regularization based on second-order derivatives. Motivated by the nature of the regularizer, we restrict the search space to the span of piecewise-linear box splines shifted on a 2D lattice. Our formulation of the infinite-dimensional problem on this search space allows us to recast it exactly as a finite-dimensional one that can be solved using standard methods in convex optimization. Since our search space is composed of continuous and piecewise-linear functions, our work presents itself as an alternative to training networks that deploy rectified linear units, which also construct models in this family. The advantages of our method are fourfold: the ability to enforce sparsity, favoring models with fewer piecewise-linear regions; the use of a rotation, scale and translation-invariant regularization; a single hyperparameter that controls the complexity of the model; and a clear model interpretability that provides a straightforward relation between the parameters and the overall learned function. We validate our framework in various experimental setups and compare it with neural networks.

***Index Terms*—Supervised learning, variational methods, sparsity, box splines, barycentric coordinates.**

I. Introduction

The primary task in supervised learning is to estimate a target function $f : \mathbb{R}^d \to \mathbb{R}$ from finitely many noisy samples $\{\mathbf{x}_m, y_m\}_{m=1}^M$, where $y_m \approx f(\mathbf{x}_m), m = 1, \ldots, M$ [1]. Since there are arbitrarily many continuous models that can fit the training data well enough, this problem is ill-posed in general. To address this issue, the learning scheme generally includes regularization and favors certain models based on prior information on the target function [2], [3].

One way to make this problem computationally tractable is to restrict the admissible solutions to a given family of parametric functions $f_{\boldsymbol{\Theta}}$, where $\boldsymbol{\Theta}$ denotes the vector of the underlying parameters. A celebrated example of this approach is deep learning, whose underlying principle is the construction of an overall map $f_{\boldsymbol{\Theta}} : \mathbb{R}^d \to \mathbb{R}$ built as a neural network via the composition of parameterized affine mappings and pointwise nonlinearities known as activation functions. The attribute "deep" refers to the high number of such module compositions (layers), which is instrumental to improve the approximation power of the network [4], [5], [6] and its generalization ability [7].

Rectified-linear-unit (ReLU) networks, in particular, have been the most prominently used in machine learning [8], [9]. These networks have spline activations of the form $x \mapsto \max(x, 0)$ which results in a continuous and piecewise-linear (CPWL) input-output relationship [4], [10]. Consequently, they can be interpreted as hierarchical splines [11], [12], [13]. Furthermore, the opposite also holds: any CPWL function can be represented by some ReLU network [14]. This leads to the conclusion that ReLU networks provide a nonlinear parameterization for the family of CPWL functions.

A more generic strategy to address supervised learning problems is to adopt a functional approach where the model is optimized over a well-suited function space [15], [16]. In this paradigm, the learning task is formalized as the minimization problem

$$\min_{f \in \mathcal{X}} \sum_{m=1}^{M} \mathrm{E}(f(\mathbf{x}_m), y_m) + \lambda \mathcal{R}(f), \tag{1}$$

where $\mathcal{X}$ is the search space, $\mathrm{E} : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}_+$ is an arbitrary convex loss function (the data-fidelity metric), $\mathcal{R} : \mathcal{X} \mapsto \mathbb{R}$ is the regularization functional, and $\lambda \in \mathbb{R}_+$ its corresponding (adjustable) weight.

A one-dimensional (1D) example of (1) is learning with second-order total-variation regularization [17], [18], which promotes sparse piecewise-linear models and leads to an alternative procedure to learn 1D CPWL functions. This problem is formulated as

$$\min_{f \in \mathrm{BV}^{(2)}(\mathbb{R})} \sum_{m=1}^{M} \mathrm{E}(f(x_m), y_m) + \lambda \mathrm{TV}^{(2)}(f), \tag{2}$$

where the search space $\mathrm{BV}^{(2)}(\mathbb{R})$ contains functions with bounded second-order total variation such that $f \in \mathrm{BV}^{(2)}(\mathbb{R}) \Leftrightarrow \mathrm{TV}^{(2)}(f) < +\infty$. In the spirit of reproducing-kernel Hilbert spaces [19], [20], there exists a representer theorem for (2) that states that the extreme points of the solution set are linear splines with the generic form

$$f(x) = b_0 + b_1 x + \sum_{k=1}^{K} a_k (x - x_k)_+, \tag{3}$$

where $\mathbf{a} = (a_k) \in \mathbb{R}^K$ and $\mathbf{b} = (b_0, b_1) \in \mathbb{R}^2$ are expansion parameters, $K < M$, and $\{x_k\}_{k=1}^K$ are adaptive (learnable) locations that are known as knots. The regularization term for these functions has the simple expression given by

$$\mathrm{TV}^{(2)}(f) = \|\mathbf{a}\|_{\ell_1}. \tag{4}$$

As a result, the original infinite-dimensional problem can be recast as a finite-dimensional one, parameterized by $K, \mathbf{a}, \mathbf{b}$, and $\{x_k\}_{k=1}^K$ [21]. The $\mathrm{TV}^{(2)}$ regularization favors simple models with few knots due to (4) and the sparsity-promoting effect of the ℓ_1-norm [22], [23].

This work was supported in part by the European Research Council (ERC Project FunLearn) under Grant 101020573 and in part by the Swiss National Science Foundation, Grant 200020_184646/1.

The authors are with the Biomedical Imaging Group, École polytechnique fédérale de Lausanne, 1015 Lausanne, Switzerland (e-mail: joaquim.campos@epfl.ch; shayan.aziznejad@epfl.ch; michael.unser@epfl.ch)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3136488, IEEE Open Journal of Signal Processing

Interestingly, one can also use $\mathrm{TV}^{(2)}$ regularization to learn the activation functions of deep neural networks. In this case, it has been shown that neural networks with linear spline activation functions of the form (3) are optimal [24], [25]. The link between functional approaches to neural networks and splines has also been observed in various works [26], [27], [28], [29], [30].

A. Contributions

Our work presents a method to learn sparse CPWL functions in 2D. It is based on three fundamental ingredients.

1) A regularization based on second-order derivatives, the Hessian-nuclear total-variation (HTV).
2) A CPWL search space spanned by linear-box-spline basis functions, which allows us to have a closed-form parametric expression for the regularization.
3) An exact discretization of the infinite-dimensional learning problem. The resulting parameterized problem has a structure that is reminiscent of the generalized LASSO [31], and therefore can be efficiently solved using known optimization algorithms. This discretization encapsulates the sparsity-promoting effect of the HTV, while it reveals the link with ℓ_1 regularization.

Our framework presents itself as an alternative to training ReLU networks for the learning of CPWL functions, with the following advantages:

1) the enforcement of sparsity, in the sense that we follow Occam's razor principle by promoting solutions with the fewest CPWL regions;
2) the use of a rotation, scale and translation-invariant regularization;
3) the reliance on a single hyperparameter—the regularization weight λ. This is in contrast with the numerous hyperparameters found in neural networks such as the choice of architecture and its components, learning rate schemes, and batch size, among others;
4) an improved model interpretability since we provide a linear parametrization for the learned CPWL mapping.

B. Related Works

One of the most widely used regularizers in image reconstruction is the Rudin-Osher-Fatemi total-variation (TV) seminorm, given by $\mathcal{R}(f) = \int_{\mathbb{R}^2} \|\boldsymbol{\nabla} f(\mathbf{x})\|_2 \, \mathrm{d}\mathbf{x}$ [32], [33], [34]. The success of TV is partly attributable to its capacity to preserve edge information. However, TV has the tendency to promote piecewise-constant solutions (vanishing first-order derivatives), which creates an undesired staircase effect. Such issue can be dealt with by the adoption of regularizers based on higher-order differentials [35], [36].

The Hessian is the second-order counterpart to the gradient operator. Accordingly, the works in [37], [38] introduce the regularizer

$$\mathcal{R}(f) = \|\mathrm{H}\{f\}\|_{*,L_1} \triangleq \int_{\mathbb{R}^2} \|\mathrm{H}\{f\}(\mathbf{x})\|_* \, \mathrm{d}\mathbf{x}, \tag{5}$$

where $\|\cdot\|_*$ is the nuclear norm—the ℓ_1-norm of the singular values of the Hessian. This regularizer preserves the desirable affine-invariance of total-variation while it alleviates the staircase effect by promoting piecewise-linear solutions.

Here, in contrast with [37], we address supervised learning rather than inverse problems. Moreover, the approach taken in [37] raises two relevant concerns. First, the regularization term (5) is inoperative for CPWL functions because the Hessian of a CPWL function is zero almost everywhere. Second, [37] resorts to a discretization of the Hessian with second-order finite differences in order to establish a discrete formulation of the problem, which leads to discretization errors. In our work, we address these two concerns by using a novel HTV seminorm as the regularization term. The proposed functional is a generalization of (5). It has been introduced in [39] to measure the complexity of neural networks. The foundation for our work is that the HTV seminorm is properly defined for CPWL functions and has an explicit closed-form formula for these mappings. This allows us to discretize our problem exactly.

C. Roadmap

In Section II, we introduce the two key mathematical elements that underlie our framework: the *Hessian-nuclear total-variation* (HTV) seminorm, along with box splines. We then define an adequate CPWL search space in Section III, where we define our HTV-regularized learning problem. In Section IV, we discretize our problem exactly and provide the algorithmic components to solve it. Finally, in Section V, we validate our framework on numerical examples.

II. Mathematical Preliminaries

A. Nuclear Norm

The nuclear norm of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is defined as

$$\|\mathbf{A}\|_* \triangleq \sum_{k=1}^{\min(m,n)} |\sigma_k| \,, \tag{6}$$

where σ_k are the singular values of $\mathbf{A}$ [40]. Its dual norm is the spectral norm, defined as $\|\mathbf{A}\|_{\mathcal{S}_\infty} \triangleq \max_k |\sigma_k|$. The nuclear norm plays a prominent role in the field of low-rank matrix recovery, due to its sparsity-promoting effect [41], [42], [43].

B. Generalized Hessian Operator

The Hessian of a twice-differentiable function $f : \mathbb{R}^2 \to \mathbb{R}$ is defined as

$$\mathrm{H}\{f\} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}. \tag{7}$$

Using this matrix, one can readily compute second-order directional derivatives. Let us recall that the second-order directional derivative of a twice-differentiable function f at $\mathbf{x}$ along a direction $\mathbf{u}$ with $\|\mathbf{u}\|_2 = 1$ is defined as

$$\mathrm{D}^2_{\mathbf{u}} f(\mathbf{x}) = \lim_{h \to 0} \frac{\mathrm{D}_{\mathbf{u}} f(\mathbf{x} + h\mathbf{u}) - \mathrm{D}_{\mathbf{u}} f(\mathbf{x})}{h}. \tag{8}$$

This quantity can be expressed as

$$\mathrm{D}^2_{\mathbf{u}} f(\mathbf{x}) = \mathbf{u}^T \mathrm{H}\{f\}(\mathbf{x}) \mathbf{u}. \tag{9}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3136488, IEEE Open Journal of Signal Processing

If the Hessian of f is symmetric, for which it suffices that the second partial derivatives be continuous (Schwarz' theorem), then its eigendecomposition has an important geometric interpretation: the two eigenvectors, $\mathbf{v}_1$ and $\mathbf{v}_2$, point in the directions in which the magnitude of the second-order directional derivative is maximal and minimal, respectively. Furthermore, the magnitudes of the directional derivatives along these directions are given by the corresponding eigenvalues, with $\left|\mathrm{D}^2_{\mathbf{v}_1} f(\mathbf{x})\right| = |\lambda_1|$, $\left|\mathrm{D}^2_{\mathbf{v}_2} f(\mathbf{x})\right| = |\lambda_2|$, respectively.

The Hessian operator can also be defined for functions that are not twice-differentiable using the notion of weak derivatives, as detailed in Appendix A.

C. Hessian-Nuclear Total Variation

In this section, we briefly recall the definition and relevant properties of the HTV seminorm (see [39] for more details). It is based on the nuclear-TV norm, which is defined for any $\mathbf{W} \in \mathcal{S}'(\mathbb{R}^2; \mathbb{R}^{2\times 2})$ as

$$\|\mathbf{W}\|_{*,\mathcal{M}} \triangleq \sup\left\{\langle \mathbf{W}, \mathbf{F}\rangle : \mathbf{F} \in \mathcal{S}(\mathbb{R}^2; \mathbb{R}^{2\times 2}), \sup_{\mathbf{x}\in\mathbb{R}^2} \|\mathbf{F}(\mathbf{x})\|_{\mathcal{S}_\infty} \leq 1\right\}. \tag{10}$$

We refer to Appendix A for more details on the matrix-valued spaces $\mathcal{S}(\mathbb{R}^2; \mathbb{R}^{2\times 2})$ and $\mathcal{S}'(\mathbb{R}^2; \mathbb{R}^{2\times 2})$. What is relevant for this paper is that, for any matrix of absolutely integrable functions $\mathbf{W} \in L_1(\mathbb{R}^2; \mathbb{R}^{2\times 2})$, we have that

$$\|\mathbf{W}(\cdot)\|_{*,\mathcal{M}} = \|\mathbf{W}(\cdot)\|_{*,L_1} = \int_{\mathbb{R}^2} \|\mathbf{W}(\mathbf{x})\|_* \, \mathrm{d}\mathbf{x}. \tag{11}$$

Furthermore, the nuclear-TV norm is defined for the Dirac-like distributions that appear in the Hessian of CPWL functions. This motivates the choice of our regularization functional given in Definition 1.

Definition 1 (Hessian-nuclear total-variation regularization)**.** *The Hessian-nuclear total-variation seminorm of $f : \mathbb{R}^2 \to \mathbb{R}$ is given by*

$$\mathrm{HTV}(f) = \|\mathrm{H}\{f\}\|_{*,\mathcal{M}}. \tag{12}$$

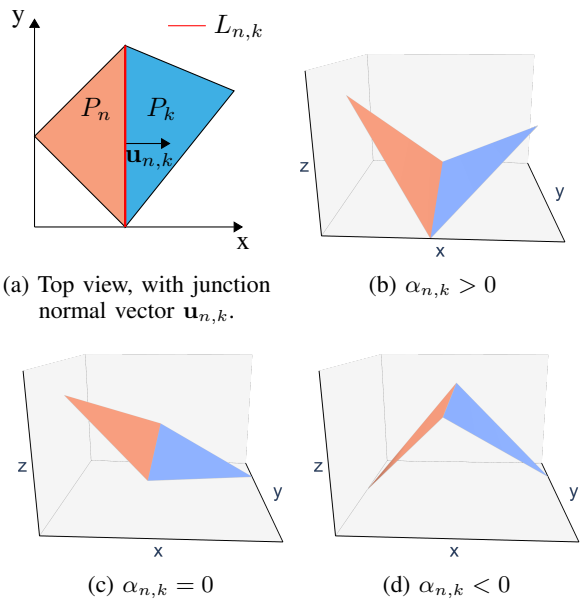
Last but not least, we mention that the HTV regularization is rotation, scale and translation-invariant [39]. Indeed, for any $f : \mathbb{R}^2 \to \mathbb{R}$ with finite HTV, any unitary matrix $\mathbf{U} \in \mathbb{R}^{2\times 2}$, scaling factor $\alpha \in \mathbb{R}$, and shift $\mathbf{x}_0 \in \mathbb{R}^2$, we have that

$$\begin{aligned} &1) \quad \mathrm{HTV}\left(f(\mathbf{U}\,\cdot)\right) = \mathrm{HTV}(f). \\ &2) \quad \mathrm{HTV}\left(f(\alpha\,\cdot)\right) = \mathrm{HTV}(f). \\ &3) \quad \mathrm{HTV}\left(f(\cdot - \mathbf{x}_0)\right) = \mathrm{HTV}(f). \end{aligned} \tag{13}$$

D. Continuous and Piecewise-Linear Functions

Definition 2 (CPWL function)**.** *A function $f : \mathbb{R}^2 \to \mathbb{R}$ is continuous and piecewise-linear if:*

1) *it is continuous $\mathbb{R}^2 \to \mathbb{R}$;*
2) *its domain $\mathbb{R}^2 = \bigsqcup_{n=1}^N P_n$ can be partitioned into a finite set of non-overlapping convex polytopes P_n over which it is affine, with $f|_{P_n}(\mathbf{x}) = \mathbf{a}_n^T\mathbf{x} + b_k$.*



(a) Top view, with junction normal vector $\mathbf{u}_{n,k}$.
(b) $\alpha_{n,k} > 0$
(c) $\alpha_{n,k} = 0$
(d) $\alpha_{n,k} < 0$

Fig. 1: $\alpha_{n,k}$ for different junctions.

The gradient of a CPWL function can easily be seen to be piecewise-constant with discontinuities at the junctions of the polytopes. For a CPWL function $f : \mathbb{R}^2 \to \mathbb{R}$, we have that

$$\boldsymbol{\nabla} f(\mathbf{x}) = \sum_{n=1}^N \mathbf{a}_n \mathbb{1}_{P_n}(\mathbf{x}), \tag{14}$$

for almost every $\mathbf{x} \in \mathbb{R}^2$. We can show that the HTV of a CPWL function $f : \mathbb{R}^2 \to \mathbb{R}$ is finite if and only if it is compactly supported up to an affine term, which means that there exists $\mathbf{a} \in \mathbb{R}^2$, $b \in \mathbb{R}$ such that $g(\mathbf{x}) = (f(\mathbf{x}) - \mathbf{a}^T\mathbf{x} - b)$ is supported over a convex and compact set $D \subseteq \mathbb{R}^2$. We call such CPWL functions "admissible". We present now a closed-form formula for the HTV of any admissible CPWL function.

Setup: We consider the partitioning of D into N polytopes $\{P_n\}_{n=1}^N$ whose gradients, $\boldsymbol{\nabla} f|_{P_n}$, are denoted by $\mathbf{a}_n$. Moreover, we gather the indices of the neighbors of a polytope P into the set $\mathrm{adj}(P)$. Each pair of neighboring polytopes P_n and P_k share a common line segment (junction) denoted by $L_{n,k} = P_n \bigcap P_k$, whose length is $\mathrm{len}(L_{n,k})$. Finally, we use the notation $\mathbf{u}_{n,k}$ for the (unit) normal vector that is perpendicular to $L_{n,k}$ (Figure 1a). The proof of Theorem 1 can be found in [39].

Theorem 1 (HTV of admissible CPWL functions)**.** *Let $f : \mathbb{R}^2 \to \mathbb{R}$ be an admissible CPWL function. Then, we have that*

$$\mathrm{HTV}(f) = \frac{1}{2}\sum_{n=1}^N \sum_{k\in\mathrm{adj}(P_n)} \left|\mathbf{u}_{n,k}^T(\mathbf{a}_k - \mathbf{a}_n)\right| \mathrm{len}(L_{n,k}) \tag{15}$$

$$= \frac{1}{2}\sum_{n=1}^N \sum_{k\in\mathrm{adj}(P_n)} \|\mathbf{a}_k - \mathbf{a}_n\|_2 \, \mathrm{len}(L_{n,k}). \tag{16}$$

Given the central character of Theorem 1 in our work, we now discuss it briefly. We first observe that $\alpha_{n,k} = \mathbf{u}_{n,k}^T(\mathbf{a}_k -$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3136488, IEEE Open Journal of Signal Processing

$\mathbf{a}_n$) is the difference of the directional derivatives inside two polytopes along the junction normal $\mathbf{u}_{n,k}$. Therefore, it measures the degree of coplanarity of the adjacent piecewise-linear regions. We illustrate how the shape of each neighboring two-polytope region relates to $\alpha_{n,k}$ in Figure 1: When the two regions are coplanar, $\alpha_{n,k} = 0$; otherwise, $|\alpha_{n,k}|$ gauges how much the slope changes in the direction $\mathbf{u}_{n,k}$ (*i.e.,* how far the function is from being affine in this region), and $\operatorname{sgn}(\alpha_{n,k})$ determines the sign of that change.

It is also interesting to compare the discrete HTV expression (15) with (4), which relates to the $\mathrm{TV}^{(2)}$ regularization in 1D . We observe that, for a CPWL function $f : \mathbb{R} \to \mathbb{R}$, $\mathrm{TV}^{(2)}(f) = \sum_{k=1}^{K} |a_k|$, where each a_k is the difference of the derivatives in two consecutive linear regions which connect at the knot position x_k. In 2D, this result is extended by considering directional derivatives and junctions instead of derivatives and knots, and by taking into account the new factor of the length of a junction. This also highlights the sparsity-promoting effect of HTV regularization: since the HTV seminorm imposes a (weighted) ℓ_1 penalty on the change of slopes of the neighbouring regions, it favors CPWL functions with few linear regions.

The necessity to restrict our framework to admissible CPWL functions is made evident by (16): the regularization only has a finite value if a function has a finite number of non-coplanar junctions, and each of these has a finite length.

Finally, we remark that, although the HTV of admissible CPWL functions has a simple closed-form expression, it requires the complete knowledge of the domain partition and the gradients in each polytope. To circumvent this, we construct a CPWL search space that is based on a uniform domain partition. This allows us to obtain a tractable formula for computing the HTV of any model in the space.

E. Box Splines

Box splines are a multivariate extension of B-splines [44]. In constrast to tensor products of 1D B-splines, they are non-separable, which makes them suitable for interpolation algorithms taylored to non-Cartesian (and often optimal) sampling lattices [45], [46], [47], [48]. They also find applications in areas such as finite-element methods [49], computer-aided design [50], and edge detection [51].

In 2D, we denote by $k_{\boldsymbol{\Xi}} : \mathbb{R}^2 \mapsto \mathbb{R}$, the box spline associated to the matrix $\boldsymbol{\Xi} = \begin{bmatrix} \boldsymbol{\xi}_1 & \cdots & \boldsymbol{\xi}_n \end{bmatrix} \in \mathbb{R}^{2\times n}$. For $n = 2$ and an invertible matrix $\boldsymbol{\Xi} \in \mathbb{R}^{2\times 2}$, $k_{\boldsymbol{\Xi}}$ is an indicator function of the form

$$k_{\boldsymbol{\Xi}}(\mathbf{x}) = \frac{1}{|\det(\boldsymbol{\Xi})|} \mathbb{1}_{S_2}(\mathbf{x}), \tag{17}$$

where $S_2 = \left\{ \boldsymbol{\Xi}\alpha : \quad \alpha \in [0,1)^2 \right\}$. For $n \geq 3$, the box spline is defined recursively as

$$k_{[\boldsymbol{\Xi} \quad \boldsymbol{\xi}]}(\mathbf{x}) = \int_0^1 k_{\boldsymbol{\Xi}}(\mathbf{x} - t\boldsymbol{\xi}) \, \mathrm{d}t. \tag{18}$$

Box splines are nonnegative functions and have a unit integral over the entire space, with $\int_{\mathbb{R}^2} k_{\boldsymbol{\Xi}}(\mathbf{x}) \, \mathrm{d}\mathbf{x} = 1$. Moreover, they are supported over the set $\left\{ \boldsymbol{\Xi}\alpha : \quad \alpha \in [0,1)^n \right\}$ and are symmetric with respect to the center of their support [52].

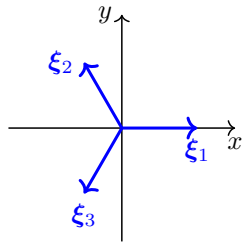


Fig. 2: Hexagonal box-spline vectors.

III. Search space

Motivated by the results of section II, we construct a search space that only contains admissible CPWL functions. More precisely, we let this space be spanned by shifts of a CPWL basis function φ of the form

$$\varphi(x_1, x_2) = [1 - \max(0, a_1, a_2) + \min(0, a_1, a_2)]_+ , \tag{19}$$

where $a_1 = (x_1 - x_2/\sqrt{3})$, $a_2 = (-2x_2/\sqrt{3})$, and $[x]_+ \triangleq \max(x, 0)$ (see Figure 3). We note that $\varphi = \frac{\sqrt{3}}{2} k_{\boldsymbol{\Xi}}$ is, in fact, a scaled hexagonal box spline [48], with the matrix $\boldsymbol{\Xi}$ given by (see Figure 2)

$$\boldsymbol{\Xi} = \begin{bmatrix} \boldsymbol{\xi}_1 & \boldsymbol{\xi}_2 & \boldsymbol{\xi}_3 \end{bmatrix} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}. \tag{20}$$

Consequently, its Fourier transform is given by

$$\widehat{\varphi}(\omega) = \frac{\sqrt{3}}{2} \prod_{n=1}^{3} \operatorname{sinc}\left(\frac{\langle \omega, \boldsymbol{\xi}_n \rangle}{2} \right). \tag{21}$$

We refer to Appendix B for the proof.

Additionally, we construct a hexagonal lattice on which we shift these basis functions. This lattice is determined by the primitive vectors $\mathbf{r}_1 = \boldsymbol{\xi}_1$ and $\mathbf{r}_2 = (-\boldsymbol{\xi}_2)$. For ease of representation, we concatenate the primitive vectors into the lattice matrix $\mathbf{R} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 \end{bmatrix}$ [53]. Then, the search space with grid size $h \in \mathbb{R}_+$ is defined as

$$\mathcal{X}_{\mathbf{R}_h}(\mathbb{R}^2) = \operatorname{span}\left(\left\{ \varphi\left(\frac{\cdot}{h} - \mathbf{R}\mathbf{k} \right) \right\}_{\mathbf{k}\in\mathbb{Z}^2} \right). \tag{22}$$

We observe that any model $f \in \mathcal{X}_{\mathbf{R}_h}(\mathbb{R}^2)$ can be expressed as

$$f(\mathbf{x}) = \sum_{\mathbf{k}\in\mathbb{Z}^2} c[\mathbf{k}] \varphi\left(\frac{\mathbf{x}}{h} - \mathbf{R}\mathbf{k} \right) \tag{23}$$

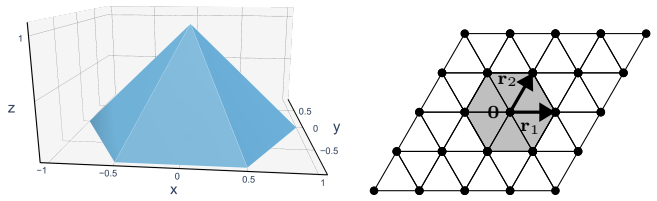
for some set of box-spline coefficients $\{c[\mathbf{k}]\}_{\mathbf{k}\in\mathbb{Z}^2}$.

Analogous to the space of cardinal linear splines [54], [55], our search space satisfies some desirable properties that are listed in Theorem 2. The proof can be found in Appendix C.

Theorem 2. *The search space $\mathcal{X}_{\mathbf{R}_h}(\mathbb{R}^2)$ satisfies the following properties.*

1) *It reproduces any affine mapping, in the sense that any function of the form $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x} + b$ can be expressed as* (23)*.*
2) *The collection $\{\varphi(\cdot/h - \mathbf{R}\mathbf{k})\}_{\mathbf{k}\in\mathbb{Z}^2}$ forms a Riesz basis for $\mathcal{X}_{\mathbf{R}_h}(\mathbb{R}^2)$. This ensures a unique and stable link between each model function and its coefficients [56].*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3136488, IEEE Open Journal of Signal Processing



(a) Basis function φ. (b) Hexagonal lattice.

Fig. 3: Principal elements of our search space.

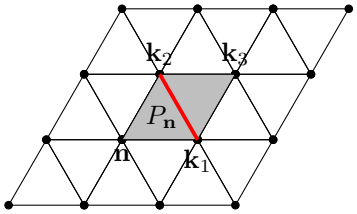


Fig. 4: Indices for two neighboring polytopes.

3) *The approximation error of our search space decays with* h^{-2} *as* $h \to 0$.
4) *The atoms satisfy the interpolatory condition*

$$\forall \mathbf{k} \in \mathbb{Z}^2 : \quad \varphi(\mathbf{R}\mathbf{k}) = \begin{cases} 1, & \mathbf{k} = 0 \\ 0, & \textit{otherwise.} \end{cases}$$

Consequently, we have that $f(h\mathbf{R}\mathbf{k}) = c[\mathbf{k}]$ *for any* $f \in \mathcal{X}_{\mathbf{R}_h}(\mathbb{R}^2)$ *and any* $\mathbf{k} \in \mathbb{Z}^2$.
5) *The basis element* φ *is refinable, in the sense that* $\varphi(\cdot/2h)$ *can be exactly represented in* $\mathcal{X}_{\mathbf{R}_h}(\mathbb{R}^2)$ *with finitely many coefficients.*

Additionally, this search space has three desirable characteristics. First, the hexagonal lattice provides an optimal packing density [57], [58], which leads to an improved approximation power for a given number of basis functions. Second, the domain of $f \in \mathcal{X}_{\mathbf{R}_h}(\mathbb{R}^2)$ is partitioned into equilateral triangles (see Figure 3b), which results in simplified computations (see Section IV). Third, Property 5 allows for efficient multiresolution algorithms.

Finally, we formalize the functional-learning problem in our search space as the minimization

$$\min_{f \in \mathcal{X}_{\mathbf{R}_h}(\mathbb{R}^2)} \left(\sum_{m=1}^{M} E\big(f(\mathbf{x}_m), y_m\big) + \lambda \mathrm{HTV}(f) \right). \tag{24}$$

where $E : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ is a strictly convex loss function (*e.g.*, $E(y, z) = (y - z)^2$ for the quadratic loss). Let us mention that our framework is compatible with any open connected bounded domain (such as Euclidean disks) [39]. Nonetheless, working with $\mathbb{R}^2$ has the advantage of learning an affine extrapolation.

IV. Discretization of the Problem

In this section, we detail the algorithm we have developed to solve problem (24). For this purpose, we first show how to efficiently evaluate (23) for a given $\mathbf{x} \in \mathbb{R}^2$, and then exactly express the HTV of any $f \in \mathcal{X}_{\mathbf{R}_h}(\mathbb{R}^2)$ in terms of its parameters.

A. Data Fidelity

Due to the finite support of the atoms (Figure 3b), we infer that, at each location $\mathbf{x} \in \mathbb{R}^2$, there are at most three basis functions that are nonzero. These active atoms are located at the vertices of the triangle to which $\mathbf{x}$ belongs. For each datapoint $\mathbf{x}_m$, let us denote its triangle by the index set $\{\mathbf{k}_{m,1}, \mathbf{k}_{m,2}, \mathbf{k}_{m,3}\}$. From this, we express $f(\mathbf{x}_m)$ as

$$\begin{aligned} f(\mathbf{x}_m) &= \sum_{n=1}^{3} c[\mathbf{k}_{m,n}] \varphi\Big(\frac{\mathbf{x}_m}{h} - \mathbf{R}\mathbf{k}_{m,n}\Big) \\ &= \mathbf{h}_m^T (c[\mathbf{k}_{m,1}], c[\mathbf{k}_{m,2}], c[\mathbf{k}_{m,3}]), \end{aligned} \tag{25}$$

where $h_{m,n} = \varphi(\mathbf{x}_m/h - \mathbf{R}\mathbf{k}_{m,n})$, $n = 1, 2, 3$.

B. Regularization

The central result of this section is presented in Theorem 3.

Theorem 3. *For any* $f \in \mathcal{X}_{\mathbf{R}_h}(\mathbb{R}^2)$ *of the form* (23)*, we have that*

$$\mathrm{HTV}(f) = \|d_1 * c\|_{1,1} + \|d_2 * c\|_{1,1} + \|d_3 * c\|_{1,1}\,, \tag{26}$$

where $\|\mathbf{A}\|_{1,1} = \|\mathrm{vec}(\mathbf{A})\|_1$ *is the sum of the absolute values of the entries of* $\mathbf{A}$*, and*

$$d_1 = \begin{bmatrix} a & -a & 0 \\ 0 & -a & a \end{bmatrix}, \; d_2 = \begin{bmatrix} a & 0 \\ -a & -a \\ 0 & a \end{bmatrix}, \; d_3 = \begin{bmatrix} -a & a \\ a & -a \end{bmatrix}, \tag{27}$$

with $a = \frac{2\sqrt{3}}{3}$.

Proof. Let Δ denote the set of triangles that form the domain partition of our search space. Adapting (16) to our search space, we have that

$$\mathrm{HTV}(f) = \frac{h}{2} \sum_{P \in \Delta} \sum_{\tilde{P} \in \mathrm{adj}(P)} \big\| \boldsymbol{\nabla} f\big|_P - \boldsymbol{\nabla} f\big|_{\tilde{P}} \big\|_2\,. \tag{28}$$

Due to the specific form of our search space, we can rewrite (28) as a summation over the lattice vertices rather than the triangles and associate three junctions to each vertex. This leads to

$$\mathrm{HTV}(f) = h \sum_{\mathbf{n} \in \mathbb{Z}^2} \sum_{k=1}^{3} \|\mathbf{a}_{\mathbf{n}_k} - \mathbf{a}_\mathbf{n}\|_2\,, \tag{29}$$

where $\mathbf{a}_\mathbf{n}$ is the gradient of the triangle $P_\mathbf{n}$ associated with the vertex $\mathbf{n}$ (Figure 4). Similarly, the vector $\mathbf{a}_{\mathbf{n}_k}$ is the gradient of the neighboring triangle that shares a border with $P_\mathbf{n}$ in the direction of $\mathbf{r}_k$, where $\mathbf{r}_1$ and $\mathbf{r}_2$ are illustrated in Figure 3b and $\mathbf{r}_3 = (-\frac{1}{2}, \frac{\sqrt{3}}{2})$. By changing the order of summation, we obtain that

$$\begin{aligned} \mathrm{HTV}(f) = h \sum_{\mathbf{n} \in \mathbb{Z}^2} \|\mathbf{a}_{\mathbf{n}_1} - \mathbf{a}_\mathbf{n}\|_2 + h \sum_{\mathbf{n} \in \mathbb{Z}^2} \|\mathbf{a}_{\mathbf{n}_2} - \mathbf{a}_\mathbf{n}\|_2 \\ + h \sum_{\mathbf{n} \in \mathbb{Z}^2} \|\mathbf{a}_{\mathbf{n}_3} - \mathbf{a}_\mathbf{n}\|_2\,. \end{aligned} \tag{30}$$

Each of the three terms of (30) can be computed via a filtering operation. Here, we just prove this for the last term in the summation and we deduce the other two using similar computations.

Using the notations in Figure 4, we write that

$$\begin{cases} \mathbf{a}_{\mathbf{n}_3}^T \mathbf{R}_h(\mathbf{k}_3 - \mathbf{k}_2) = c[\mathbf{k}_3] - c[\mathbf{k}_2] \\ \mathbf{a}_{\mathbf{n}_3}^T \mathbf{R}_h(\mathbf{k}_3 - \mathbf{k}_1) = c[\mathbf{k}_3] - c[\mathbf{k}_1] \\ \mathbf{a}_{\mathbf{n}}^T \mathbf{R}_h(\mathbf{k}_1 - \mathbf{n}) = c[\mathbf{k}_1] - c[\mathbf{n}] \\ \mathbf{a}_{\mathbf{n}}^T \mathbf{R}_h(\mathbf{k}_2 - \mathbf{n}) = c[\mathbf{k}_2] - c[\mathbf{n}] \end{cases} \Leftrightarrow$$

$$\begin{cases} \mathbf{R}_h^T \mathbf{a}_{\mathbf{n}_3} = \begin{bmatrix} c[\mathbf{k}_3] - c[\mathbf{k}_2] \\ c[\mathbf{k}_3] - c[\mathbf{k}_1] \end{bmatrix} \\ \mathbf{R}_h^T \mathbf{a}_{\mathbf{n}} = \begin{bmatrix} c[\mathbf{k}_1] - c[\mathbf{n}] \\ c[\mathbf{k}_2] - c[\mathbf{n}]] \end{bmatrix}, \end{cases} \tag{31}$$

where $\mathbf{R}_h = h \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 \end{bmatrix}$ is the lattice matrix. Combining these equations, we obtain that

$$\mathbf{R}_h^T(\mathbf{a}_{\mathbf{n}_3} - \mathbf{a}_{\mathbf{n}}) = (c[\mathbf{n}] - c[\mathbf{k}_1] - c[\mathbf{k}_2] + c[\mathbf{k}_3])\mathbf{1}, \tag{32}$$

where $\mathbf{1} = (1, 1)$. The application of $\left(\mathbf{R}_h^{-1}\right)^T$ to both sides of (32) leads to

$$(\mathbf{a}_{\mathbf{n}_3} - \mathbf{a}_n) = (1, -1, -1, 1)^T \mathbf{z} \left(\mathbf{R}_h^{-1}\right)^T \mathbf{1}, \tag{33}$$

where $\mathbf{z} = (c[\mathbf{n}], c[\mathbf{k}_1], c[\mathbf{k}_2], c[\mathbf{k}_3])$. Using the homogeneity of the ℓ_2-norm, we verify that

$$\begin{aligned} \|\mathbf{a}_{\mathbf{n}_3} - \mathbf{a}_{\mathbf{n}}\|_2 &= \left|(1, -1, -1, 1)^T \mathbf{z}\right| \left\| \left(\mathbf{R}_h^{-1}\right)^T \mathbf{1} \right\|_2 \\ &= \frac{2\sqrt{3}}{3h} \left|(1, -1, -1, 1)^T \mathbf{z}\right|. \end{aligned} \tag{34}$$

By plugging in $\mathbf{k}_1 = \mathbf{n} + (1, 0)$, $\mathbf{k}_2 = \mathbf{n} + (0, 1)$ and $\mathbf{k}_3 = \mathbf{n} + (1, 1)$, we express the last term in (30) as

$$\begin{aligned} h \sum_{\mathbf{n} \in \mathbb{Z}^2} \|\mathbf{a}_{\mathbf{n}_3} - \mathbf{a}_{\mathbf{n}}\|_2 = \frac{2\sqrt{3}}{3} \sum_{\mathbf{n} \in \mathbb{Z}^2} \big| & c[\mathbf{n}] - c[\mathbf{n} + (1, 0)] - \\ & c[\mathbf{n} + (0, 1)] + c[\mathbf{n} + (1, 1)] \big| \\ = \|d_3 * c\|_{1,1}. & \end{aligned} \tag{35}$$

■

Theorem 3 provides a simple algorithm to evaluate $\mathrm{HTV}(f)$ in terms of three convolutions, whose filters are depicted in Figure 5. We also remark that, for any admissible CPWL model f, the outputs of the digital filters $d_n, n = 1, 2, 3$, are zero outside of a compact domain. This in effect allows us to consider an equivalent finite lattice to represent f in practice.

C. Generalized LASSO

We now merge the results of sections III, IV-A, and IV-B to derive an exact finite-dimensional discretization of Problem (24). We consider a finite lattice of square size $(N \times N)$ in such a way that all training data are contained in it. The lattice coefficients are grouped into the vector $\mathbf{c} \in \mathbb{R}^{N^2}$ which is a (row-wise) vectorization of the 2D array $c[\mathbf{k}], \mathbf{k} \in \Omega$, where

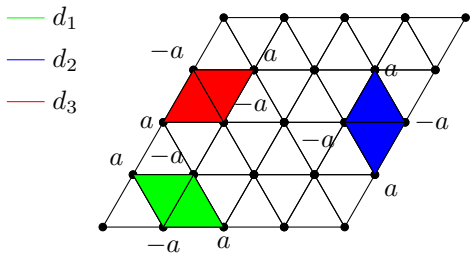


Fig. 5: Convolutional filters; $a = \frac{2\sqrt{3}}{3}$.

$\Omega \subset \mathbb{Z}^2$ is the set of lattice indices. Consequently, we define the regularization matrix $\mathbf{L} \in \mathbb{R}^{3N^2 \times N^2}$ as

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_1 \\ \mathbf{L}_2 \\ \mathbf{L}_3 \end{bmatrix}, \tag{36}$$

where $\mathbf{L}_n$ is a Toeplitz-like matrix associated to the 2D digital filter d_n such that, for $n = 1, 2, 3$, $\mathbf{L}_n \mathbf{c}$ is the vectorized version of $(d_n * c)[\mathbf{k}], \mathbf{k} \in \Omega$. Further, we define the forward matrix $\mathbf{H} \in \mathbb{R}^{M \times N^2}$ such that its mth row corresponds to the datapoint $\mathbf{x}_m$, with $f(\mathbf{x}_m) = [\mathbf{H}\mathbf{c}]_m$ for $m = 1, \ldots, M$. Using these, we restate (24) as the finite-dimensional minimization

$$\arg \min_{\mathbf{c} \in \mathbb{R}^{N^2}} \sum_{m=1}^{M} E\left([\mathbf{H}\mathbf{c}]_m, y_m\right) + \lambda \|\mathbf{L}\mathbf{c}\|_1. \tag{37}$$

Ultimately, the finite-dimensional problem (37) has the composite structure of the generalized LASSO [31] which can be solved efficiently using known convex optimization solvers (*e.g.,* ADMM or its variants [59], [60], [61]). We denote the corresponding solution by $\mathbf{c}_0$.

The discrete formulation (37) also highlights the sparsity-promoting effect of the HTV regularization, due to the presence of the ℓ_1 penalty in (37). Consequently, we expect to learn models with few linear regions. In order to find a sparser solution, we use the simplex algorithm [62], [63] to solve the minimization

$$\arg \min_{\mathbf{c} \in \mathbb{R}^{N^2}} \|\mathbf{L}\mathbf{c}\|_1, \ s.t. \ \mathbf{H}\mathbf{c} = \mathbf{H}\mathbf{c}_0. \tag{38}$$

This post-processing step is known to provide an extreme point of the solution set of (37) [21] which, in our case, often leads to a sparser CPWL mapping.

D. Discussion

In an attempt to generalize our framework, we could also consider learning higher-order splines. Indeed, the HTV is fully compatible with these smooth functions, and we can even define different flavors of it by choosing other Schatten-norms [39]. The main challenge would be to obtain a parametric expression (akin to (26)) for the HTV of a given element in the restricted search space. This is crucial for obtaining an exact discretization of the continuous-domain learning problem.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3136488, IEEE Open Journal of Signal Processing

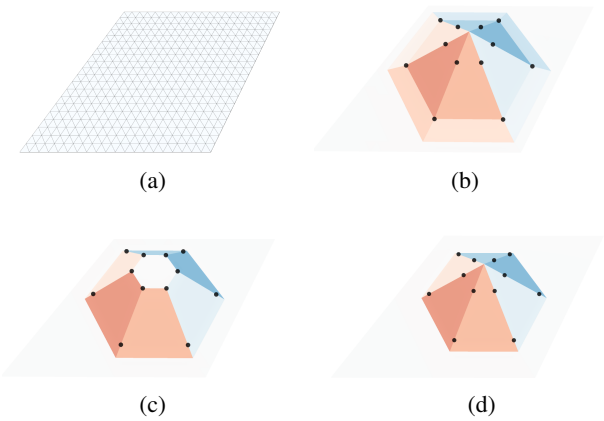


Fig. 6: Solutions of the minimum-norm interpolation problem.

V. Experiments

In this section, we demonstrate the advantages of our pipeline by comparing it to other existing learning methods. The Python code to reproduce all the experimental results is available on Github[1].

A. Mininum-Norm Interpolation

We demonstrate the sparsity-promoting effect of the HTV regularizer in a controlled environment in which we sample $M = 12$ points from a pyramid function f_{pyr} whose vertices are positioned on the lattice. This ensures that the target function can be represented exactly in our search space. To isolate the effect of the regularization, we use simplex solver to find

$$\arg \min_{f \in \mathcal{X}_{\mathbf{R}_h}(\mathbb{R}^2)} \text{HTV}(f), \quad s.t.\ f(\mathbf{x}_m) = f_{\text{pyr}}(\mathbf{x}_m), \\ m = 1, \dots, M \tag{39}$$

by recasting it as a discrete minimization problem of the form (38).

In Figure 6, we show the results of successive experiments where we use a lattice of size (20×20) (a total number of 421 parameters) with zero boundary conditions. We chose a colormap based on the triangle normals so that co-planarities can be identified. Due to randomness in the implementation of the algorithm, we obtained different solutions of (38). They all resulted in the same minimal $\text{HTV}(f)$. We observe that the algorithm leads to sparse solutions in all cases, with few faces. Indeed, from a search space which can model functions with hundreds of faces (Figure 6a), we reached solutions with just 12 (6b), 7 (6c), and 6 (6b) faces, respectively.

B. Data Fitting

In this experiment, we tackle a data-fitting problem and compare three approaches.

[1]https://github.com/joaquimcampos/HTV-Learn

1) **Ours**, using HTV regularization and a CPWL search space (22).
2) **ReLU neural networks**, which also construct CPWL models.
3) **Radial basis functions** with Gaussian kernels—a classical approach in supervised learning [64], [65], [66].

The dataset consists of samples from a CPWL function with noisy labels. More precisely, the labels are of the form

$$y_m = h(\mathbf{x}_m) + \epsilon, \tag{40}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and h is the CPWL function shown in Figures 7a and 7f. We use 200 datapoints and set $\sigma = \frac{1}{20} \|f\|_{L_\infty}$. Note that the model cannot be represented exactly in our search space since the data points do not fall on the lattice; however, the error can be mitigated by a sufficient reduction of the stepsize of the grid.

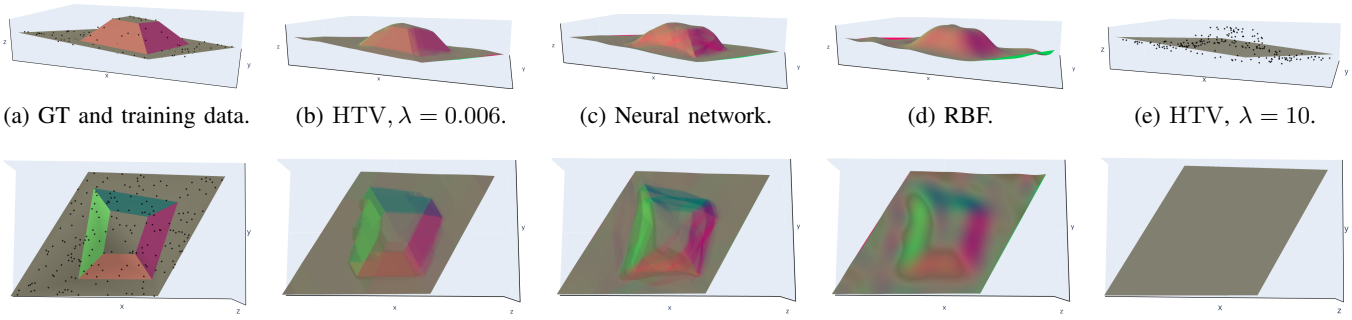
The setup is as follows: for the data-fidelity term in (37), we use the quadratic loss $E(y, z) = (y - z)^2$. For the ReLU network, we use a fully connected architecture with 4 hidden layers, each with 256 hidden neurons. The total number of parameters of the neural network is 198401. We train the neural network for 500 epochs using an Adam optimizer [67] with a batch size of 10 and weight decay. The initial learning rate is set to 10^{-3} and is decreased by 10 at epochs 375 and 425. For the HTV, we use a lattice size of size (64×64), giving a total of 4225 parameters. In all methods, we tune the corresponding hyperparameter on a validation set (regularization weight λ for the HTV and radial-basis function [RBF], kernel size γ for the RBF, and weight-decay parameter μ for the neural network) to have a fair comparison. To assess sparsity, we sample the learned neural network and RBF models in the position of the lattice vertices and vectorize these values (we denote the resulting vector by $\mathbf{c}$), as done for our method. Finally, for all methods, we compute the percentage of non-negligible "changes of slope" as $\frac{\|\mathbf{L}\mathbf{c} > \epsilon\|_0}{3N^2} \cdot 100$, where $\epsilon = 10^{-4}$ and $3N^2$ is the number of rows of $\mathbf{L}$.

The results are shown in Table I and Figure 7, along with the ground-thruth (GT). We observe that the HTV model performs significantly better than the radial-basis functions and on par with the neural network. Furthermore, as seen in the last column of the table and from the Figures, the HTV leads to a much sparser result. Moreover, we observe that the level of sparsity for the HTV can be controlled with the regularization weight (higher leads to sparser results). In the extreme case $\lambda \to +\infty$, the model should converge to the least-squares linear approximation of the training data. The very high regularization weight $\lambda = 10$ allows us to verify this in practice. Indeed, the resulting model is linear and the data-fitting error is precisely the same as the one obtained with a least-squares fit.

C. Real Dataset

In this section, we benchmark the three methods of the experiment of section V-B on a (non-CPWL) facial dataset. This dataset is a 2D height map $f : \mathbb{R}^2 \to \mathbb{R}$ that we construct

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3136488, IEEE Open Journal of Signal Processing



(a) GT and training data. (b) HTV, $\lambda = 0.006$. (c) Neural network. (d) RBF. (e) HTV, $\lambda = 10$.

(f) GT and training data. (g) HTV, $\lambda = 0.006$. (h) Neural network. (i) RBF. (j) HTV, $\lambda = 10$.

Fig. 7: HTV regularization *vs.* ReLU neural network *vs.* radial basis functions.

Model	Hyperparameters	Test MSE	Sparsity
HTV	$\lambda = 2 \times 10^{-3}$	$\mathbf{3.6 \times 10^{-5}}$	**27**%
HTV	$\lambda = 4 \times 10^{-3}$	3.8×10^{-5}	19%
HTV	$\lambda = 6 \times 10^{-3}$	4.4×10^{-5}	16%
HTV	$\lambda = 10$	3.1×10^{-3}	00%
ReLU	$\mu = 1 \times 10^{-6}$	$\mathbf{3.7 \times 10^{-5}}$	**63**%
RBF	$\lambda = 0.08, \gamma = 7$	5.5×10^{-5}	88%

TABLE I: Test MSE and sparsity of each method in the data-fitting example.

Model	Hyperparameters	Test MSE	Sparsity
HTV	$\lambda = 2 \times 10^{-3}$	$\mathbf{3.0 \times 10^{-6}}$	**10**%
HTV	$\lambda = 7 \times 10^{-3}$	4.8×10^{-6}	8%
HTV	$\lambda = 5 \times 10^{-2}$	1.9×10^{-5}	6%
ReLU	$\mu = 1 \times 10^{-6}$	$\mathbf{5.1 \times 10^{-6}}$	**12**%
RBF	$\lambda = 10^{-4}, \gamma = 50$	3.2×10^{-6}	31%
RBF	$\lambda = 10^{-2}, \gamma = 50$	3.4×10^{-6}	24%

TABLE II: Test MSE and sparsity of each method in the face dataset.

by cutting a 3D face model[2] (Figure 8a). We then sample 8000 data points for training (Figure 8b).

Relative to Section V-B, the setup has the following differences: for the HTV, we use a lattice of size (194×194) (38025 parameters) and skip the simplex post-processing step; for the neural network, we incorporate one additional hidden layer (264193 parameters), increase the number of epochs to 2000 and the batch size to 100, and, lastly, decrease the initial learning rate at epochs 1750 and 1900.

The results are shown in Table II and Figure 8. The HTV achieved the lowest test mean-square error (MSE) on par with the RBF which is expected to perform well due to the high density of datapoints and the absence of noise. Regarding the effect of the regularization, we again observe that, for the HTV, increasing it results in a model with fewer faces. In the case of the RBF, the solutions present ringing artifacts, especially in a low-regularization regime. Finally, we remark that the neural network constructs a coarse approximation of the data.

VI. Conclusion

We have introduced a method to solve two-dimensional learning problems regularized with Hessian-nuclear total-variation (HTV) seminorm. The starting point of our work has been the observation that the HTV of (admissible) continuous and piecewise-linear (CPWL) functions has a closed-form expression. Its computation, however, requires knowledge of the gradient and boundaries of each partition. To circumvent this drawback, we have formulated the problem in a search space consisting of shifts of CPWL box-splines in a lattice. By doing so, we are able to evaluate any model in the search space, as well as compute its HTV, from the values at the lattice points (model parameters). In particular, we showed that the latter can be computed with a three-filter convolutional structure; this allows us to discretize the problem exactly and to recast it in the form of the generalized least-absolute shrinkage-and-selection operator. Finally, we have demonstrated the sparsity-promoting effect of our framework via numerical examples where we have compared its performance with ReLU neural networks and radial-basis functions.

[2]https://www.turbosquid.com/3d-models/3d-male-head-model-1357522

Appendix

A. Generalized Hessian Operator

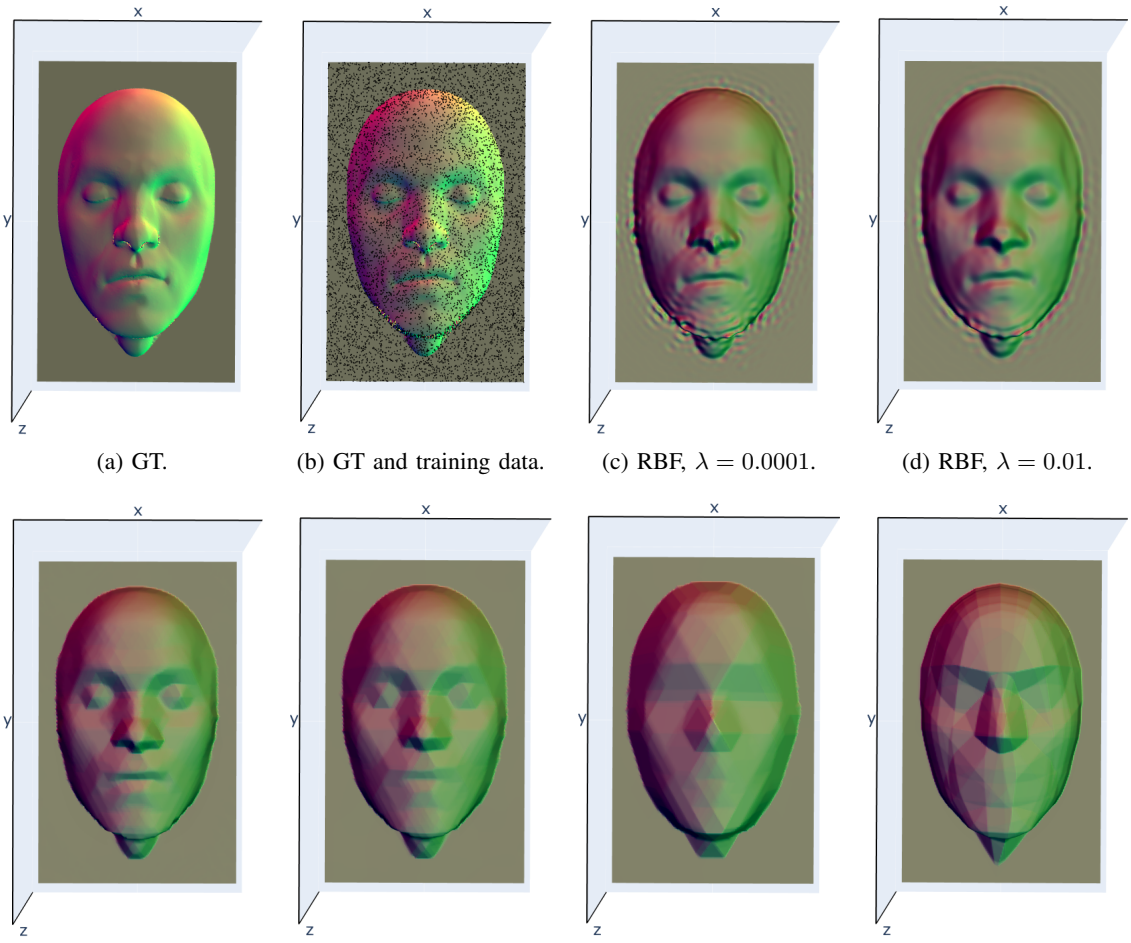
Let $\mathcal{S}(\mathbb{R}^2)$ be the Schwartz space of smooth and rapidly decaying functions over $\mathbb{R}^2$. Its topological dual, denoted by $\mathcal{S}'(\mathbb{R}^2)$, is the space of tempered distributions. An element $w \in \mathcal{S}'(\mathbb{R}^2)$ is a linear functional whose action for any $\varphi \in \mathcal{S}(\mathbb{R}^2)$ is given by the duality product $\varphi \mapsto \langle w, \varphi \rangle$.

For any $w \in \mathcal{S}'(\mathbb{R}^2)$, the weak partial derivative $\frac{\partial}{\partial x_k} : \mathcal{S}'(\mathbb{R}^2) \to \mathcal{S}'(\mathbb{R}^2)$ is a continuous linear map whose action is given by

$$\left\langle \frac{\partial w}{\partial x_k}, \varphi \right\rangle = \left\langle w, -\frac{\partial \varphi}{\partial x_k} \right\rangle, \quad \forall \varphi \in \mathcal{S}(\mathbb{R}^2),\ k = 1, 2, \quad (41)$$

where the partial derivative on the right-hand side is in the usual sense (with the limit definition) $\frac{\partial}{\partial x_k} : \mathcal{S}(\mathbb{R}^2) \to \mathcal{S}(\mathbb{R}^2)$. We refer to [68, Section 3.3.2] for more details on extensions by duality.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/OJSP.2021.3136488, IEEE Open Journal of Signal Processing



(a) GT. (b) GT and training data. (c) RBF, $\lambda = 0.0001$. (d) RBF, $\lambda = 0.01$.

(e) HTV, $\lambda = 0.002$. (f) HTV, $\lambda = 0.007$. (g) HTV, $\lambda = 0.05$. (h) Neural network.

Fig. 8: HTV regularization *vs.* ReLU neural network *vs.* radial basis functions.

To define the generalized Hessian operator, we first need to extend the two spaces $\mathcal{S}(\mathbb{R}^2)$ and $\mathcal{S}'(\mathbb{R}^2)$ for matrix-valued functions. Concretely, any $\mathbf{F} \in \mathcal{S}(\mathbb{R}^2; \mathbb{R}^{2\times 2})$ is of the form

$$\mathbf{F} = \begin{bmatrix} f_{1,1} & f_{1,2} \\ f_{2,1} & f_{2,2} \end{bmatrix}, \ f_{m,n} \in \mathcal{S}(\mathbb{R}^2). \tag{42}$$

Similarly, any $\mathbf{W} \in \mathcal{S}'(\mathbb{R}^2; \mathbb{R}^{2\times 2})$ is of the form

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} \\ w_{2,1} & w_{2,2} \end{bmatrix}, \ w_{m,n} \in \mathcal{S}'(\mathbb{R}^2). \tag{43}$$

Consequently, the duality product associated with these two spaces is the sum of the entrywise duality products

$$\langle \mathbf{W}, \mathbf{F} \rangle = \sum_{m=1}^{2} \sum_{n=1}^{2} \langle w_{m,n}, f_{m,n} \rangle. \tag{44}$$

Finally, the generalized Hessian operator is denoted by $\mathrm{H} : \mathcal{S}'(\mathbb{R}^2) \to \mathcal{S}'(\mathbb{R}^2; \mathbb{R}^{2\times 2})$ and is defined for any $f \in \mathcal{S}'(\mathbb{R}^2)$ as

$$\mathrm{H}\{f\} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}. \tag{45}$$

B. Atoms of the Search Space

Let $\alpha = 2/\sqrt{3}$ and $\boldsymbol{\Xi} \in \mathbb{R}^{2\times 3}$, the matrix given in (20). It follows from Definition (18) that

$$\begin{aligned} k_{\boldsymbol{\Xi}}(\mathbf{x}) &= \alpha \int_0^1 \mathbb{1}_{S_2}(\mathbf{x} - t\boldsymbol{\xi}_3), \in \mathrm{d}t \\ &= \alpha \ \mathrm{supp}\left([0,1] \cap \{t : \ \mathbf{x} - t\boldsymbol{\xi}_3 \in S_2\}\right), \end{aligned} \tag{46}$$

where $\mathrm{supp}(B)$ is the length of the interval $B \in \mathbb{R}$ and $S_2 = \{t_1\boldsymbol{\xi}_1 + t_2\boldsymbol{\xi}_2 : \ t_1, t_2 \in [0,1)\}$. By expressing $\mathbf{x}$ in the basis $\{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2\}$, with $\mathbf{x} = a_1\boldsymbol{\xi}_1 + a_2\boldsymbol{\xi}_2$, $a_1, a_2 \in \mathbb{R}$, and using the relation $(-\boldsymbol{\xi}_3) = \boldsymbol{\xi}_1 + \boldsymbol{\xi}_2$, we deduce that

$$\begin{aligned} k_{h\boldsymbol{\Xi}}(\mathbf{x}) &= \alpha \ \mathrm{supp}\left([0,1] \cap \{t : \ (a_1 + t)\boldsymbol{\xi}_1 + \right. \\ &\qquad \left. (a_2 + t)\boldsymbol{\xi}_2 \in S_2\}\right) \\ &= \alpha \ \mathrm{supp}\left(\{t : \ 0 \le t \le 1, \ 0 \le a_1 + t \le 1, \right. \\ &\qquad \left. 0 \le a_2 + t \le 1\}\right) \\ &= \alpha \left[\min(1, 1 - a_1, 1 - a_2) - \max(0, -a_1, -a_2)\right]_+ \\ &= \alpha \left[1 - \max(0, a_1, a_2) + \min(0, a_1, a_2)\right]_+ . \end{aligned} \tag{47}$$

Finally, dividing both sides by α, we reach the desired result.

C. Proof of Theorem 2

We prove the properties for a unit grid size ($h = 1$), without any loss of generality. To do so, we rely on the Fourier-domain characterization of a generic box spline [69, Proposition (17)]. Specifying it for the case of the hexagonal box spline whose vectors are given in (20), and using $\boldsymbol{\xi}_1 + \boldsymbol{\xi}_2 + \boldsymbol{\xi}_3 = 0$ and the relation $\varphi = \sqrt{3}/2\, k_{\boldsymbol{\Xi}}$ (Appendix B), we get that

$$\widehat{\varphi}(\omega) = \frac{\sqrt{3}}{2} \prod_{n=1}^{3} \operatorname{sinc}\left(\frac{\langle \omega, \boldsymbol{\xi}_n \rangle}{2}\right), \tag{48}$$

where $\operatorname{sinc}(x) = \frac{\sin(x)}{x}$.

Item 1 (Reproduction of affine mappings): We begin by proving that $\{\varphi(\cdot - \mathbf{R}\mathbf{k})\}_{\mathbf{k}\in\mathbb{Z}^2}$ satisfies the partition-of-unity property

$$\sum_{\mathbf{k}\in\mathbb{Z}^2} \varphi(\mathbf{x} - \mathbf{R}\mathbf{k}) = 1, \qquad \forall \mathbf{x} \in \mathbb{R}^2. \tag{49}$$

This condition implies that the search space is able to reproduce any constant function.

Let $\tilde{\mathbf{R}}$ be the lattice matrix expressed as $\tilde{\mathbf{R}} = \begin{bmatrix}\boldsymbol{\xi}_1 & \boldsymbol{\xi}_2\end{bmatrix}$. One can readily verify that $\sum_{\mathbf{k}\in\mathbb{Z}^2} \varphi(\mathbf{x} - \mathbf{R}\mathbf{k}) = \sum_{\mathbf{k}\in\mathbb{Z}^2} \varphi(\mathbf{x} - \tilde{\mathbf{R}}\mathbf{k})$. From the Poisson-sum formula for lattices [53], the partition-of-unity property holds if and only if, for any $\mathbf{k} \in \mathbb{Z}^2$, we have that

$$\frac{1}{\left|\det(\tilde{\mathbf{R}})\right|} \widehat{\varphi}(2\pi\tilde{\mathbf{R}}^{-T}\mathbf{k}) = \begin{cases} 1, & \mathbf{k} = \mathbf{0} \\ 0, & \mathbf{k} \neq \mathbf{0}. \end{cases} \tag{50}$$

Evaluating the Fourier transform (48) at the selected locations and using $\left|\det(\tilde{\mathbf{R}})\right| = \frac{\sqrt{3}}{2}$, we infer that

$$\begin{aligned} \frac{1}{\left|\det(\tilde{\mathbf{R}})\right|} \widehat{\varphi}(2\pi\tilde{\mathbf{R}}^{-T}\mathbf{k}) &= \prod_{n=1}^{3} \operatorname{sinc}\left(\pi\langle \tilde{\mathbf{R}}^{-T}\mathbf{k}, \boldsymbol{\xi}_n\rangle\right) \\ &= \prod_{n=1}^{3} \operatorname{sinc}\left(\pi\langle \mathbf{k}, \tilde{\mathbf{R}}^{-1}\boldsymbol{\xi}_n\rangle\right). \end{aligned} \tag{51}$$

We then observe that $\tilde{\mathbf{R}}^{-1}\boldsymbol{\xi}_1 = (1, 0)$, $\tilde{\mathbf{R}}^{-1}\boldsymbol{\xi}_2 = (0, 1)$, and $\tilde{\mathbf{R}}^{-1}\boldsymbol{\xi}_3 = (-1, -1)$. This results in

$$\begin{aligned} \frac{1}{\left|\det(\tilde{\mathbf{R}})\right|} \widehat{\varphi}(2\pi\tilde{\mathbf{R}}^{-T}\mathbf{k}) &= \operatorname{sinc}\left(\pi(k_1 + k_2)\right) \prod_{n=1}^{2} \operatorname{sinc}(\pi k_n) \\ &= \begin{cases} 1, & \mathbf{k} = \mathbf{0} \\ 0, & \mathbf{k} \neq \mathbf{0}, \end{cases} \end{aligned} \tag{52}$$

where, in the last equality, we have used that $\operatorname{sinc}(\pi k) = \delta[k]$ for any $k \in \mathbb{Z}$.

Now, we show that the search space can approximate any linear function. Following the Strang-Fix conditions [46], [70], we just need to prove that

$$\nabla\widehat{\varphi}(2\pi\tilde{\mathbf{R}}^{-T}\mathbf{k}) = \mathbf{0}, \ \forall \mathbf{k} \in \mathbb{Z}^2 \setminus \{\mathbf{0}\}. \tag{53}$$

Using the product rule for differentiation, we observe that

$$\nabla\widehat{\varphi}(\omega) = \frac{\sqrt{3}}{2} \sum_{n=1}^{3} \boldsymbol{\nabla} \operatorname{sinc}\left(\frac{\langle \omega, \boldsymbol{\xi}_n \rangle}{2}\right) \prod_{m\neq n} \operatorname{sinc}\left(\frac{\langle \omega, \boldsymbol{\xi}_m \rangle}{2}\right). \tag{54}$$

Evaluating this expression at $\omega = 2\pi\tilde{\mathbf{R}}^{-T}\mathbf{k}$ and defining $\boldsymbol{\beta}_{n,\mathbf{k}} = \left(\sqrt{3}/2\right) \boldsymbol{\nabla} \operatorname{sinc}(\pi\langle \tilde{\mathbf{R}}^{-T}\mathbf{k}, \boldsymbol{\xi}_n\rangle)$, we obtain that

$$\nabla\widehat{\varphi}(2\pi\tilde{\mathbf{R}}^{-T}\mathbf{k}) = \sum_{n=1}^{3} \boldsymbol{\beta}_{n,\mathbf{k}} \prod_{m\neq n} \operatorname{sinc}\left(\pi\langle \tilde{\mathbf{R}}^{-T}\mathbf{k}, \boldsymbol{\xi}_m\rangle\right). \tag{55}$$

Then, we use that $\operatorname{sinc}\left(\pi\langle \tilde{\mathbf{R}}^{-T}\mathbf{k}, \boldsymbol{\xi}_m\rangle\right) = \operatorname{sinc}\left(\pi\langle \mathbf{k}, \tilde{\mathbf{R}}^{-1}\boldsymbol{\xi}_m\rangle\right)$ to deduce that

$$\begin{aligned} \boldsymbol{\nabla}\widehat{\varphi}(2\pi\tilde{\mathbf{R}}^{-T}\mathbf{k}) = \boldsymbol{\beta}_{1,\mathbf{k}} &\operatorname{sinc}(\pi k_2) \operatorname{sinc}(\pi(k_1 + k_2)) \\ &+ \boldsymbol{\beta}_{2,\mathbf{k}} \operatorname{sinc}(\pi k_1) \operatorname{sinc}(\pi(k_1 + k_2)) \\ &+ \boldsymbol{\beta}_{3,\mathbf{k}} \operatorname{sinc}(\pi k_1) \operatorname{sinc}(\pi k_2). \end{aligned} \tag{56}$$

Finally, since $\operatorname{sinc}(\pi k) = \delta[k]$ for any $k \in \mathbb{Z}$, all terms in (56) vanish for $\mathbf{k} \in \mathbb{Z} \setminus \{\mathbf{0}\}$.

Item 2 (Riesz basis): The collection $\{\varphi(\cdot - \mathbf{R}\mathbf{k})\}_{\mathbf{k}\in\mathbb{Z}^2}$ is a Riesz basis if there exist $\lambda_{\min} > 0$ and $\lambda_{\max} < +\infty$ such that, for any sequence $c \in \ell_2(\mathbb{Z})$, we have that

$$\lambda_{\min} \|c\|_2^2 \leq \left\| \sum_{\mathbf{k}\in\mathbb{Z}^2} c[\mathbf{k}]\varphi(\cdot - \mathbf{R}\mathbf{k}) \right\|_{L_2}^2 \leq \lambda_{\max} \|c\|_2^2. \tag{57}$$

To show that (57) is valid for the collection of our shifted search-space atoms, we use Fourier-based conditions in the spirit of [53] and [56]. This leads to the bounds

$$\begin{aligned} \lambda_{\min} &= \min_{[0,2\pi)^2} \frac{1}{|\det(\mathbf{R})|} \sum_{\mathbf{k}\in\mathbb{Z}^2} \left|\widehat{\varphi}(\mathbf{R}^{-T}(\omega + 2\pi\mathbf{k}))\right|^2, \\ \lambda_{\max} &= \max_{[0,2\pi)^2} \frac{1}{|\det(\mathbf{R})|} \sum_{\mathbf{k}\in\mathbb{Z}^2} \left|\widehat{\varphi}(\mathbf{R}^{-T}(\omega + 2\pi\mathbf{k}))\right|^2. \end{aligned} \tag{58}$$

To obtain a more tractable expression for the summation on the right-hand side of (58), we set $\mathbf{x} = \mathbf{0}$ in the Poisson-sum formula for lattices [53] and deduce that

$$\sum_{\mathbf{k}\in\mathbb{Z}^2} f(\mathbf{R}\mathbf{k}) = \frac{1}{|\det(\mathbf{R})|} \sum_{\mathbf{k}\in\mathbb{Z}^2} \widehat{f}(2\pi\mathbf{R}^{-T}\mathbf{k}). \tag{59}$$

Then, we consider the function $f(\tau) = c_{\varphi\varphi}(\tau)\mathrm{e}^{-\mathrm{j}\langle \mathbf{R}^{-T}\omega_0, \tau\rangle}$, where $c_{\varphi\varphi}(\tau) = \langle \varphi(\cdot - \tau), \varphi\rangle$, which results in

$$\begin{aligned} \sum_{\mathbf{k}\in\mathbb{Z}^2} \langle \varphi(\cdot - \mathbf{R}\mathbf{k}), \varphi\rangle \mathrm{e}^{-\mathrm{j}\langle \omega, \mathbf{k}\rangle} = \\ \frac{1}{|\det(\mathbf{R})|} \sum_{\mathbf{k}\in\mathbb{Z}^2} \left|\widehat{\varphi}\big(\mathbf{R}^{-T}(\omega + 2\pi\mathbf{k})\big)\right|^2, \end{aligned} \tag{60}$$

where we have used that $\widehat{c}_{\varphi\varphi}(\omega) = \left|\widehat{\varphi}(\omega)\right|^2$ and taken advantage of the modulation property of the Fourier transform.

Due to the fact that φ is finitely supported, the summation on the left-hand side of (60) contains only 7 nonzero terms: 1 term corresponding to the energy of the atom and 6 others corresponding to the inner product with overlapping replicas (Figure 9). Therefore, (60) can be expanded as

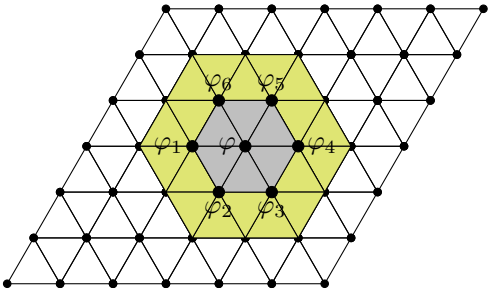


Fig. 9: Overlapping replicas of φ. The non-white region indicates the union of the support of the replicas.

$$\begin{aligned}\sum_{\mathbf{k}\in\mathbb{Z}^2}\langle\varphi(\cdot-\mathbf{Rk}),\varphi\rangle\mathrm{e}^{-\mathrm{j}\langle\omega,\mathbf{k}\rangle} = \\ \langle\varphi(\cdot-\mathbf{r}_1),\varphi\rangle\mathrm{e}^{-\mathrm{j}\omega_1}+\langle\varphi(\cdot+\mathbf{r}_1),\varphi\rangle\mathrm{e}^{\mathrm{j}\omega_1} \\ +\langle\varphi(\cdot-\mathbf{r}_2),\varphi\rangle\mathrm{e}^{-\mathrm{j}\omega_2}+\langle\varphi(\cdot+\mathbf{r}_2),\varphi\rangle\mathrm{e}^{\mathrm{j}\omega_2} \\ +\langle\varphi(\cdot+\mathbf{r}_1-\mathbf{r}_2),\varphi\rangle\mathrm{e}^{-\mathrm{j}(\omega_2-\omega_1)} \\ +\langle\varphi(\cdot-\mathbf{r}_1+\mathbf{r}_2),\varphi\rangle\mathrm{e}^{\mathrm{j}(\omega_2-\omega_1)}+\|\varphi\|_{L_2}^2\,. \end{aligned} \tag{61}$$

We remark that the pairs of conjugate exponentials in (61) do arise due to the symmetry in the location of the replicas. By simple computations, we deduce that $\|\varphi\|_{L_2}^2 = \frac{\sqrt{3}}{4}$ and $\langle\varphi_n,\varphi\rangle = \frac{\sqrt{3}}{12}$ for any of the replicas φ_n, $n=1,\dots,6$. (Due to symmetries, the inner products are all equal.) Combining (61) with (58) and (60), we conclude that

$$\begin{aligned}\lambda_{\min} &= \frac{\sqrt{3}}{12}\min_{[0,2\pi)^2}\left(3+\cos(\omega_1)+\cos(\omega_2)+\cos(-\omega_1+\omega_2)\right) \\ &= \frac{\sqrt{3}}{8} > 0, \\ \lambda_{\max} &= \frac{\sqrt{3}}{12}\max_{[0,2\pi)^2}\left(3+\cos(\omega_1)+\cos(\omega_2)+\cos(-\omega_1+\omega_2)\right) \\ &= \frac{\sqrt{3}}{2} < +\infty, \end{aligned} \tag{62}$$

which completes the proof.

Item 3 (Order of approximation): From Items 1 and 2, we know that the collection of the search-space atoms $\{\varphi(\cdot-\mathbf{Rk})\}_{\mathbf{k}\in\mathbb{Z}^2}$ forms a Riesz basis and reproduces first-degree polynomials. Hence, it satisfies the first-order Strang-Fix conditions [71, Theorem 2.2.]. It follows that

$$\left\|f-\mathrm{Proj}_{\mathcal{X}_{\mathbf{R}_\mathrm{h}}}\{f\}\right\|_{L_2} = \mathcal{O}(h^{-2}),\ h\to 0 \tag{63}$$

for any sufficiently smooth function $f:\mathbb{R}^2\to\mathbb{R}$ [70].

Item 4 (Interpolatory atoms): Evaluating the partition of unity at $\mathbf{x}=\mathbf{Rk}'$, we have that

$$\sum_{\mathbf{k}\in\mathbb{Z}^2}\varphi(\mathbf{R}(\mathbf{k}'-\mathbf{k})) = \varphi(\mathbf{0})+\sum_{\mathbf{k}\neq\mathbf{k}'}\varphi(\mathbf{R}(\mathbf{k}'-\mathbf{k})) = 1. \tag{64}$$

Since $\varphi(\mathbf{0})=1$ and $\varphi(\mathbf{x})\geq 0,\ \forall\,\mathbf{x}\in\mathbb{R}^2$, it follows that

$$\forall\,\mathbf{k}\in\mathbb{Z}^2:\quad \varphi(\mathbf{Rk}) = \begin{cases}1, & \mathbf{k}=0\\ 0, & \text{Otherwise.}\end{cases} \tag{65}$$

Item 5 (Refinable search space): We want to show that there exists a refinability filter $h\in\ell_2(\mathbb{R}^2)$ such that

$$\varphi\Big(\frac{\mathbf{x}}{2}\Big) = \sum_{\mathbf{k}\in\mathbb{Z}^2}h[\mathbf{k}]\varphi(\mathbf{x}-\tilde{\mathbf{R}}\mathbf{k}), \tag{66}$$

where $\tilde{\mathbf{R}} = \begin{bmatrix}\boldsymbol{\xi}_1 & \boldsymbol{\xi}_2\end{bmatrix}$. In the Fourier domain, this condition is equivalent to

$$2^2\widehat{\varphi}(2\omega) = \mathrm{H}(\mathrm{e}^{\mathrm{j}\tilde{\mathbf{R}}^T\omega})\widehat{\varphi}(\omega). \tag{67}$$

Computing $2^2\widehat{\varphi}(2\omega)/\widehat{\varphi}(\omega)$, we deduce that

$$\begin{aligned}\mathrm{H}(\mathrm{e}^{\mathrm{j}\tilde{\mathbf{R}}^T\omega}) &= \frac{4\widehat{\varphi}(2\omega)}{\widehat{\varphi}(\omega)} = 4\prod_{n=1}^{3}\frac{\mathrm{sinc}(\langle\omega,\boldsymbol{\xi}_n\rangle)}{\mathrm{sinc}\left(\frac{\langle\omega,\boldsymbol{\xi}_n\rangle}{2}\right)} \\ &= 2\prod_{n=1}^{3}\frac{\sin(\langle\omega,\boldsymbol{\xi}_n\rangle)}{\sin\left(\frac{\langle\omega,\boldsymbol{\xi}_n\rangle}{2}\right)} = 4\prod_{n=1}^{3}\cos\left(\frac{\langle\omega,\boldsymbol{\xi}_n\rangle}{2}\right), \end{aligned} \tag{68}$$

where we have used the identity $\sin(x) = 2\cos(x/2)\sin(x/2)$. Observing that $\mathrm{H}(\mathrm{e}^{\mathrm{j}\omega}) = \mathrm{H}(\mathrm{e}^{\mathrm{j}\tilde{\mathbf{R}}^T(\tilde{\mathbf{R}}^{-T}\omega)})$, we get that

$$\begin{aligned}\mathrm{H}(\mathrm{e}^{\mathrm{j}\omega}) &= 4\prod_{n=1}^{3}\cos\left(\frac{\langle\omega,\tilde{\mathbf{R}}^{-1}\boldsymbol{\xi}_n\rangle}{2}\right) \\ &= 4\cos\left(\frac{w_1}{2}\right)\cos\left(\frac{w_2}{2}\right)\cos\left(\frac{w_1+w_2}{2}\right) \\ &= \frac{1}{2}\Big(1+\mathrm{e}^{-\mathrm{j}w_1}\Big)\Big(1+\mathrm{e}^{-\mathrm{j}w_2}\Big)\Big(1+\mathrm{e}^{\mathrm{j}(w_1+w_2)}\Big), \end{aligned} \tag{69}$$

where $\langle\tilde{\mathbf{R}}^{-T}\omega,\boldsymbol{\xi}_n\rangle = \langle\omega,\tilde{\mathbf{R}}^{-1}\boldsymbol{\xi}_n\rangle$ and $\boldsymbol{\xi}_3 = (-\boldsymbol{\xi}_1-\boldsymbol{\xi}_2)$. Taking the inverse Fourier transform of (69), we write that

$$\begin{aligned}h[k_1,k_2] = \delta[k_1,k_2]+\frac{1}{2}\big(&\delta[k_1-1,k_2]+\delta[k_1+1,k_2] \\ &+\delta[k_1,k_2-1]+\delta[k_1,k_2+1] \\ &+\delta[k_1-1,k_2-1]+\delta[k_1+1,k_2+1]\big). \end{aligned} \tag{70}$$

Finally, replacing (70) in (66), and again using that $\boldsymbol{\xi}_3 = (-\boldsymbol{\xi}_1-\boldsymbol{\xi}_2)$, we obtain that

$$\begin{aligned}\varphi\Big(\frac{\mathbf{x}}{2}\Big) = \frac{1}{2}\sum_{\mathbf{k}\in\{0,1\}^3}\varphi(\mathbf{x}-\boldsymbol{\Xi}\mathbf{k}) = \\ +\varphi(\mathbf{x})+\frac{1}{2}\Big(&\varphi(\mathbf{x}-\boldsymbol{\xi}_1)+\varphi(\mathbf{x}-\boldsymbol{\xi}_2) \\ &+\varphi(\mathbf{x}-\boldsymbol{\xi}_3)+\varphi(\mathbf{x}-\boldsymbol{\xi}_1-\boldsymbol{\xi}_2) \\ &+\varphi(\mathbf{x}-\boldsymbol{\xi}_1-\boldsymbol{\xi}_3)+\varphi(\mathbf{x}-\boldsymbol{\xi}_2-\boldsymbol{\xi}_3)\Big). \end{aligned} \tag{71}$$

REFERENCES

[1] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer, 2006.
[2] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, ser. Adaptive Computation and Machine Learning Series. MIT Press, 2012.
[3] F. Girosi, M. Jones, and T. Poggio, "Regularization Theory and Neural Networks Architectures," *Neural Computation*, vol. 7, no. 2, pp. 219–269, Mar. 1995.
[4] G. F. Montufar, R. Pascanu, K. Cho, and Y. Bengio, "On the Number of Linear Regions of Deep Neural Networks," in *Proceedings of the 27th Conference on Advances in Neural Information Processing Systems*, vol. 27, Montréal, Canada, Dec. 2014.

[5] R. Eldan and O. Shamir, "The Power of Depth for Feedforward Neural Networks," in *Proceedings of the 29th Conference on Learning Theory*, vol. 49, New York, USA, Jun. 2016, pp. 907–940.

[6] H. N. Mhaskar and T. Poggio, "Deep vs. Shallow Networks: An Approximation Theory Perspective," *Analysis and Applications*, vol. 14, no. 06, pp. 829–848, Nov. 2016.

[7] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao, "Why and When can Deep—But Not Shallow—Networks Avoid the Curse of Dimensionality: A Review," *International Journal of Automation and Computing*, vol. 14, no. 5, pp. 503–519, Oct. 2017.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[9] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, vol. 15, Florida, USA, Apr. 2011, pp. 315–323.

[10] R. Pascanu, G. Montufar, and Y. Bengio, "On the Number of Response Regions of Deep Feed Forward Networks With Piece-Wise Linear Activations," *arXiv preprint arXiv:1312.6098*, Feb. 2014.

[11] T. Poggio, L. Rosasco, A. Shashua, N. Cohen, and F. Anselmi, "Notes on Hierarchical Splines, DLCNs and i-theory," Center for Brains, Minds and Machines (CBMM), Memo 37, 2015.

[12] R. Balestriero and R. Baraniuk, "A Spline Theory of Deep Learning," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, Stockholm, Sweden, Jul. 2018, pp. 374–383.

[13] R. Balestriero and R. G. Baraniuk, "Mad Max: Affine Spline Insights Into Deep Learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 704–727, May 2021.

[14] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, "Understanding Deep Neural Networks with Rectified Linear Units," in *Proceedings of the 6th International Conference on Learning Representations*, Vancouver, Canada, Apr. 30.

[15] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2001.

[16] M. Unser, "A Unifying Representer Theorem for Inverse Problems and Machine Learning," *Foundations of Computational Mathematics*, Sep. 2020.

[17] T. Debarre, Q. Denoyelle, M. Unser, and J. Fageot, "Sparsest Continuous Piecewise-Linear Representation of Data," *arXiv preprint arXiv:2003.10112*, Mar. 2020.

[18] M. Unser, J. Fageot, and J. P. Ward, "Splines are Universal Solutions of Linear Inverse Problems with Generalized-TV regularization," *SIAM Review*, vol. 59, no. 4, pp. 769–793, Jan. 2017.

[19] N. Aronszajn, "Theory of Reproducing Kernels," *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–337, Mar. 1950.

[20] B. Schölkopf, R. Herbrich, and A. J. Smola, "A Generalized Representer Theorem," in *Computational Learning Theory*. Springer Berlin Heidelberg, 2001, vol. 2111, pp. 416–426.

[21] H. Gupta, J. Fageot, and M. Unser, "Continuous-Domain Solutions of Linear Inverse Problems with Tikhonov *versus* Generalized TV Regularization," *IEEE Transactions on Signal Processing*, vol. 66, no. 17, pp. 4670–4684, Sep. 2018.

[22] D. L. Donoho, "For Most Large Underdetermined Systems of Linear Equations the Minimal ℓ_1-Norm Solution Is Also the Sparsest Solution," *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[23] R. Tibshirani, "Regression Shrinkage and Selection via the LASSO: A Retrospective," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 73, no. 3, pp. 273–282, Jun. 2011.

[24] M. Unser, "A Representer Theorem for Deep Neural Networks," *Journal of Machine Learning Research*, vol. 20, no. 110, pp. 1–30, Jan. 2019.

[25] S. Aziznejad, H. Gupta, J. Campos, and M. Unser, "Deep Neural Networks with Trainable Activations and Controlled Lipschitz Constant," *IEEE Transactions on Signal Processing*, vol. 68, pp. 4688–4699, Aug. 2020.

[26] R. Parhi and R. D. Nowak, "The Role of Neural Network Activation Functions," *IEEE Signal Processing Letters*, vol. 27, pp. 1779–1783, 2020.

[27] ——, "Neural Networks, Ridge Splines, and TV Regularization in the Radon Domain," *arXiv preprint arXiv:2006.05626*, Jun. 2020.

[28] T. Ergen and M. Pilanci, "Revealing the Structure of Deep Neural Networks via Convex Duality," in *38th International Conference on Machine Learning*, Jul. 2021.

[29] ——, "Convex Geometry and Duality of Over-parameterized Neural Networks," *arXiv preprint arXiv:2002.11219*, Apr. 2020.

[30] P. Bohra, J. Campos, H. Gupta, S. Aziznejad, and M. Unser, "Learning Activation Functions in Deep (Spline) Neural Networks," *IEEE Open Journal of Signal Processing*, vol. 1, pp. 295–309, Nov. 2020.

[31] R. J. Tibshirani and J. Taylor, "The Solution Path of the Generalized LASSO," *The Annals of Statistics*, vol. 39, no. 3, pp. 1335–1371, Jun. 2011.

[32] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, Nov. 1992.

[33] P. Getreuer, "Rudin-Osher-Fatemi Total Variation Denoising using Split Bregman," *Image Processing On Line*, vol. 2, pp. 74–95, May 2012.

[34] J. M. Bioucas-Dias, M. A. T. Figueiredo, and J. P. Oliveira, "Total Variation-Based Image Deconvolution: A Majorization-Minimization Approach," in *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 2, Toulouse, France, May 2006, pp. II–861–II–864.

[35] J. Yuan, C. Schnörr, and G. Steidl, "Total-Variation Based Piecewise Affine Regularization," in *Scale Space and Variational Methods in Computer Vision*. Springer Berlin Heidelberg, 2009, vol. 5567, pp. 552–564.

[36] M. Lysaker and X.-C. Tai, "Iterative Image Restoration Combining Total Variation Minimization and a Second-Order Functional," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 5–18, Jan. 2006.

[37] S. Lefkimmiatis, J. P. Ward, and M. Unser, "Hessian Schatten-Norm Regularization for Linear Inverse Problems," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1873–1888, May 2013.

[38] S. Lefkimmiatis, A. Bourquard, and M. Unser, "Hessian-Based Norm Regularization for Image Restoration with Biomedical Applications," *IEEE Transactions on Image Processing*, vol. 21, no. 3, pp. 983–995, Mar. 2012.

[39] S. Aziznejad, J. Campos, and M. Unser, "Measuring Complexity of Learning Schemes Using Hessian-Schatten Total-Variation," *arXiv preprint arXiv:2112.06209*, Dec. 2021.

[40] R. Bhatia, *Matrix Analysis*, ser. Graduate Texts in Mathematics. Springer New York, 1997, vol. 169.

[41] E. J. Candès and B. Recht, "Exact Matrix Completion via Convex Optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, p. 717, Apr. 2009.

[42] D. Gross, "Recovering Low-Rank Matrices from Few Coefficients in Any Basis," *IEEE Transactions on Information Theory*, vol. 57, no. 3, pp. 1548–1566, Mar. 2011.

[43] B. Recht, "A simpler approach to matrix completion," *Journal of Machine Learning Research*, vol. 12, no. 104, pp. 3413–3430, Dec. 2011.

[44] A. Entezari, M. Nilchian, and M. Unser, "A Box Spline Calculus for the Discretization of Computed Tomography Reconstruction Problems," *IEEE Transactions on Medical Imaging*, vol. 31, no. 8, pp. 1532–1541, Aug. 2012.

[45] A. Entezari, "Optimal sampling lattices and trivariate box splines," Ph.D. dissertation, Simon Fraser University, Jul. 2007.

[46] A. Entezari, D. Van De Ville, and T. Möller, "Practical Box Splines for Reconstruction on the Body Centered Cubic Lattice," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 2, pp. 313–328, Mar. 2008.

[47] H. Kunsch, E. Agrell, and F. Hamprecht, "Optimal Lattices for Sampling," *IEEE Transactions on Information Theory*, vol. 51, no. 2, pp. 634–647, Feb. 2005.

[48] L. Condat and D. Van De Ville, "Three-Directional Box-Splines: Characterization and Efficient Evaluation," *IEEE Signal Processing Letters*, vol. 13, no. 7, pp. 417–420, Jul. 2006.

[49] W. Dahmen and C. A. Micchelli, "On the Optimal Approximation Rates for Criss-Cross Finite Element Spaces," *Journal of Computational and Applied Mathematics*, vol. 10, no. 3, pp. 255–273, Jun. 1984.

[50] H. Prautzsch and W. Boehm, "Box Splines," in *Handbook of Computer Aided Geometric Design*. Amsterdam: North-Holland, 2002, pp. 255–282.

[51] W. Guo and M.-J. Lai, "Box Spline Wavelet Frames for Image Edge Analysis," *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1553–1578, Jan. 2013.

[52] H. Prautzsch, W. Boehm, and M. Paluszny, "Box Splines," in *Bézier and B-Spline Techniques*. Springer Berlin Heidelberg, 2002, pp. 239–258.

[53] D. Van De Ville, T. Blu, M. Unser, W. Philips, I. Lemahieu, and R. Van de Walle, "Hex-Splines: A Novel Spline Family for Hexagonal Lattices," *IEEE Transactions on Image Processing*, vol. 13, no. 6, pp. 758–772, Jun. 2004.

[54] M. Unser, "Splines: A Perfect Fit for Signal and Image Processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22–38, Nov. 1999.

[55] ——, "Sampling—50 Years After Shannon," *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, Apr. 2000.
[56] A. Aldroubi and M. Unser, "Sampling Procedures in Function Spaces and Asymptotic Equivalence with Shannon's Sampling Theory," *Numerical Functional Analysis and Optimization*, vol. 15, no. 1-2, pp. 1–21, Jan. 1994.
[57] H.-C. Chang and L.-C. Wang, "A Simple Proof of Thue's Theorem on Circle Packing," *arXiv preprint arXiv:1009.4322*, Sep. 2010.
[58] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, ser. Grundlehren Der Mathematischen Wissenschaften. Springer New York, 1999, vol. 290.
[59] S. Boyd, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, Jan. 2011.
[60] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proceedings of the 24th Conference on Advances in Neural Information Processing Systems*, vol. 24, Granada, Spain, Dec. 2011.
[61] N. Parikh and S. Boyd, "Proximal Algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123–231, Jan. 2014.
[62] G. B. Dantzig, A. Orden, and P. S. Wolfe, *Notes on Linear Programming: Part I: The Generalized Simplex Method for Minimizing a Linear Form Under Linear Inequality Restraints*. RAND Corporation, 1954.
[63] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, ser. Operations Research & Management Science. Springer New York, 2008, vol. 116.
[64] F. Girosi, M. Jones, and T. Poggio, "Priors Stabilizers and Basis Functions: From Regularization to Radial, Tensor and Additive Splines," Massachusetts Institute of Technology, USA, Artificial Intelligence Memo 1430, 1993.
[65] I. Steinwart, D. Hush, and C. Scovel, "An Explicit Description of the Reproducing Kernel Hilbert Spaces of Gaussian RBF Kernels," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4635–4643, Oct. 2006.
[66] S. Aziznejad and M. Unser, "Multikernel Regression with Sparsity Constraint," *SIAM Journal on Mathematics of Data Science*, vol. 3, no. 1, pp. 201–224, Feb. 2021.
[67] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, Jan. 2017.
[68] M. Unser and P. Tafti, *An Introduction to Sparse Stochastic Processes*. Cambridge University Press, 2014.
[69] C. De Boor, K. Höllig, and S. D. Riemenschneider, *Box Splines*. Springer, 2011.
[70] G. Strang and G. Fix, "A Fourier Analysis of the Finite Element Variational Method," in *Constructive Aspects of Functional Analysis*. Springer Berlin Heidelberg, 2011, pp. 793–840.
[71] M. Unser, "Approximation Power of Biorthogonal Wavelet Expansions," *IEEE Transactions on Signal Processing*, vol. 44, no. 3, pp. 519–527, Mar. 1996.