

# Learning sub-patterns in piecewise continuous functions

**Journal Article****Author(s):**

[Kratsios, Anastasis](#) ; Zamanlooy, Behnoosh

**Publication date:**

2022-04-01

**Permanent link:**

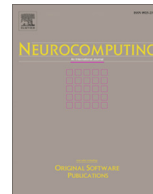
<https://doi.org/10.3929/ethz-b-000530954>

**Rights / license:**

[Creative Commons Attribution 4.0 International](#)

**Originally published in:**

Neurocomputing 480, <https://doi.org/10.1016/j.neucom.2022.01.036>



# Learning sub-patterns in piecewise continuous functions <sup>☆</sup>

Anastasis Kratsios <sup>a,b,\*</sup>, Behnoosh Zamanlooy <sup>c,\*</sup>

<sup>a</sup>ETH Zürich, Department of Mathematics, Rämistrasse 101, 8092 Zürich, Switzerland

<sup>b</sup>University of Basel, Department of Mathematics and Informatics, Spiegelgasse 1, 4051 Basel, Switzerland

<sup>c</sup>University of Zürich, Department of Informatics, Binzmühlestrasse 14, 8050 Zürich, Switzerland



## ARTICLE INFO

### Article history:

Received 22 May 2021

Revised 13 October 2021

Accepted 14 January 2022

Available online 21 January 2022

### 2020 MSC:

68T07

49J53

05A18

65Y05

68W20

91G15

### Keywords:

Piecewise continuous functions

Universal approximation

Discontinuous feedforward networks

Discontinuous functions

Deep zero-sets

Set-valued universal approximation

Quantitative finance

## ABSTRACT

Most stochastic gradient descent algorithms can optimize neural networks that are sub-differentiable in their parameters; however, this implies that the neural network's activation function must exhibit a degree of continuity which limits the neural network model's uniform approximation capacity to continuous functions. This paper focuses on the case where the discontinuities arise from distinct sub-patterns, each defined on different parts of the input space. We propose a new discontinuous deep neural network model trainable via a decoupled two-step procedure that avoids passing gradient updates through the network's only and strategically placed, discontinuous unit. We provide approximation guarantees for our architecture in the space of bounded continuous functions and universal approximation guarantees in the space of piecewise continuous functions which we introduced herein. We present a novel semi-supervised two-step training procedure for our discontinuous deep learning model, tailored to its structure, and we provide theoretical support for its effectiveness. The performance of our model and trained with the propose procedure is evaluated experimentally on both real-world financial datasets and synthetic datasets.

© 2022 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Since their introduction in [1], neural networks have led to numerous advances across various scientific areas. These include, mathematical finance in [2–4], computer vision and neuroimaging in [5,6], signal processing in [7,8], and climate change modeling in [9]. From the theoretical vantage point, these methods' success lies in the harmony between their expressivity [10–13], the training algorithms which can efficiently leverage this expressibility [14–17] and, the implicit inductive bias of deep neural model trained with these methods [18,19].

This paper probes the first two aspects when faced with the task of learning piecewise continuous functions. We first identify approximation-theoretic limitations to commonly deployed feed-

forward neural networks (FFNNs); i.e.: with continuous activation functions, and then fill this gap with a new deep neural model (PCNNs) together with a randomized and parallelizable training meta-algorithm that exploits the PCNN's structure.

The description of the problem, and our results, begins by revisiting the classical *universal approximation theorems* [20–22]. Briefly, these classical universal approximation results state that, when a phenomenon is governed by some continuous target function  $f$ , then FFNNs with continuous function  $\sigma$  can control the worst-case approximation of error to arbitrary precision. If  $f$  is discontinuous, as is for instance the case in many signal processing or mathematical finance [23,24] situations then, the uniform limit theorem from classical topology [25] guarantees that the worst-case approximation error of  $f$  by FFNNs cannot be controlled; nevertheless, the average error incurred by approximating  $f$  by FFNNs can be [11,26,27]. Essentially, this means that if  $f$  is discontinuous then when approximating it by an FFNN there must be a “small” portion of the (non-empty) input space  $X \subseteq \mathbb{R}^d$  whereon the approximation can become arbitrarily poor.

Most of A. Kratsios's work on this paper was completed during his time at ETH Zürich  
\* Corresponding authors.

E-mail addresses: [anastasis.kratsios@unibas.ch](mailto:anastasis.kratsios@unibas.ch) (A. Kratsios), [bzamanlooy@ifi.uzh.ch](mailto:bzamanlooy@ifi.uzh.ch) (B. Zamanlooy).

If  $f : X \rightarrow \mathbb{R}^D$  is a *piecewise continuous function*, by which we mean that it can be represented as:

$$f = \sum_{n=1}^N I_{K_n} f_n; \tag{1}$$

for some integer  $N$ , continuous functions  $f_n : \mathbb{R}^d \rightarrow \mathbb{R}^D$  and, some (non-empty) compact subsets  $K_n \subseteq X$  then, the regions where the approximation of  $f$  by FFNN is poor corresponds precisely to the regions where the *parts*  $\{K_n\}_{n=1}^N$  interface (whenever the *subpatterns*  $\{f_n\}_{n=1}^N$  miss-match thereon).

In principle, the existence result of [28] implies that guarantees for uniform approximation should be possible with a deep neural model and one such likely candidate are deep feedforward with discontinuous activation functions. However, these types of networks are not compatible with most commonly used (stochastic) gradient descent-type algorithms. A few methods for training such models are available. For instance, [29] describes a heuristic approach, but its empirical performance and theoretical guarantees are not explored. In [30] the author proposes a linear programming approach to training shallow feedforward networks with threshold activation function whose hidden weights and biases are fixed. This avoids back-propagating through the non-differentiable threshold activation function; however, the method is specific to their shallow architecture. Similarly, [31] considers an extreme learning machine approach by randomizing all but the network’s final linear layer, which reduces the training task to a classical linear regression problem. However, this approach’s provided approximation results are strictly weaker than the known guarantees for classical feedforward networks with a continuous activation function obtained, as are derived for example in [32].

Our proposed solution to overcoming the problem that: worst-case *universal approximation* by FFNNs is limited to continuous functions, begins by acknowledging that approximating an arbitrary discontinuous (or even integrable) function  $f$  is an *unstructured approximation* problem; whereas approximating a piecewise continuous function is a *structured approximation problem*; i.e. there is additional structure and it should be encoded into the machine learning model in order to achieve competent predictive performance. Typical examples of structured approximation problems include assimilating graph structure into the learning model [33,34], manifold-valued neural networks [35–38], encoded symmetries into the trained model [39–41], or encode inevitability [42–45]. Here, the relevant structure is given by the (continuous) subpatterns  $\{f_n\}_{n=1}^N$  and (non-empty compact) parts  $\{K_n\}_{n=1}^N$  representing the function  $f$  in (1).

Our proposed deep neural models, the PCNNs, reflects this structure by approximately parameterized the sub-patterns  $\{f_n\}_{n=1}^N$  by in dependant FFNNs, approximately parameterizing the parts  $\{K_n\}_{n=1}^N$  by zero-sets of FFNNs, and the combining these independent parts via a single discontinuous unit defined shortly. An instance of our architecture is illustrated graphically in Fig. 2. Since each FFNN component of our architecture is in dependant from one another and are only regrouped at their final outputs by the discontinuous unit, illustrated in red in Fig. 2, then we can leverage this structure to decouple the training of each component of our PCNN model and then re-combine them when producing predictions. Thus, our decoupled training procedure allows effectively avoids passing any gradients through the discontinuous unit. Thus, our architecture enjoys the expressivity of a discontinuous units (guaranteed by our universal approximation theorems for piecewise continuous functions in Theorems 3.0.1, 3.0.2, and 3.2.1) while being pragmatically trainable (illustrated in by our numerical experiments and previewed in Fig. 1).

Fig. 1 provides a concrete visual motivation of our results and approach. It illustrates the challenge of learning a piecewise continuous functions with two parts (in grey and orange) by an FFNN with ReLU activation function 2 hidden layers and 100 neurons in each layer (in purple) and a comparable PCNN model (green). After 3000 iterations of ADAM algorithm of [14], the training stabilizes at a mean absolute error (MAE) of about 0.1477. In contrast the PCNNs are capable produced an MAE of 0.986. see Fig. 3.

### 1.0.1. Organization of paper

Section 2 contains the background material required in the formulation of the paper’s main results. Section 3 includes the paper’s main results. These include qualitative universal approximation guarantees for the PCNN as a whole, as well as, for each of its individual components, quantitative approximation guarantees for the PCNN given a partition, as well as a result showing that the PCNN are dense in a significantly larger space than the FFNNs are, for the uniform distance. We then introduce a randomized algorithm that exploits the PCNN’s structure to train it. Theoretical guarantees surrounding this algorithm are also proven. Lastly, in Section 4 we validate our theoretical claims by training the PCNN architecture using our proposed meta-algorithm to generate predictions from various real-world financial and synthetic datasets. The model’s performance is benchmarked against comparable deep neural models trained using conventional training algorithms.

## 2. Preliminaries

Let us fix some notation. The following notation is used and maintained throughout the paper. The space of continuous functions from  $X$  to  $\mathbb{R}^D$  is denoted by  $C(\mathbb{R}^d, \mathbb{R}^D)$ . When  $D = 1$ , we follow the convention of denoting  $C(\mathbb{R}^d, \mathbb{R})$  by  $C(\mathbb{R}^d)$ . Throughout this paper, we fix a continuous *activation function*  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ . We denote the set of all feedforward networks from  $\mathbb{R}^d$  to  $\mathbb{R}^D$  by  $\mathbf{NN}_{d,D}^\sigma$  and we denote  $\sigma_{\text{sigmoid}}(x) \triangleq \frac{e^x}{1+e^x}$ .

We denote the set of positive integers by  $\mathbb{N}^+$ . The cardinality (or size) of a set  $A$  is denoted  $\#A$ . In this manuscript,  $d, D$  will always be non-negative integers. Moreover,  $X$  will always denote a non-empty compact subset of  $\mathbb{R}^d$ .

### 2.1. The PCNN model

Before reviewing the relevant background for the formulation of our results, we first define the PCNN architecture.

**Definition 2.1.** [PCNN] PCNN is a function  $\hat{f} : X \rightarrow \mathbb{R}^D$  with representation

$$\hat{f} = \sum_{n=1}^N \hat{f}_n(x) I_{\hat{K}_n}(x),$$

where the partition  $\{\hat{K}_n\}_{n=1}^N$  is given by the “deep zero-sets” defined as:

$$\hat{K}_n = \{x \in X : 1 - I_{(\gamma,1]} \circ \sigma_{\text{sigmoid}}(\hat{c}(x)_n) = 0\}, \tag{2}$$

and where the “sub-patterns”  $\hat{f}_1, \dots, \hat{f}_N$  are FFNNs in  $\mathbf{NN}_{d,N}^\sigma, \hat{c} \in \mathbf{NN}_{d,N}^\sigma$

and  $0 < \gamma \leq 1, N \in \mathbb{N}_+$ . The set of PCNNs is denoted by  $\mathbf{PCNN}_{d,D}^\sigma$ .

Each  $\hat{f}_1, \dots, \hat{f}_N$  is called a sub-pattern of the PCNN  $\hat{f}$ .

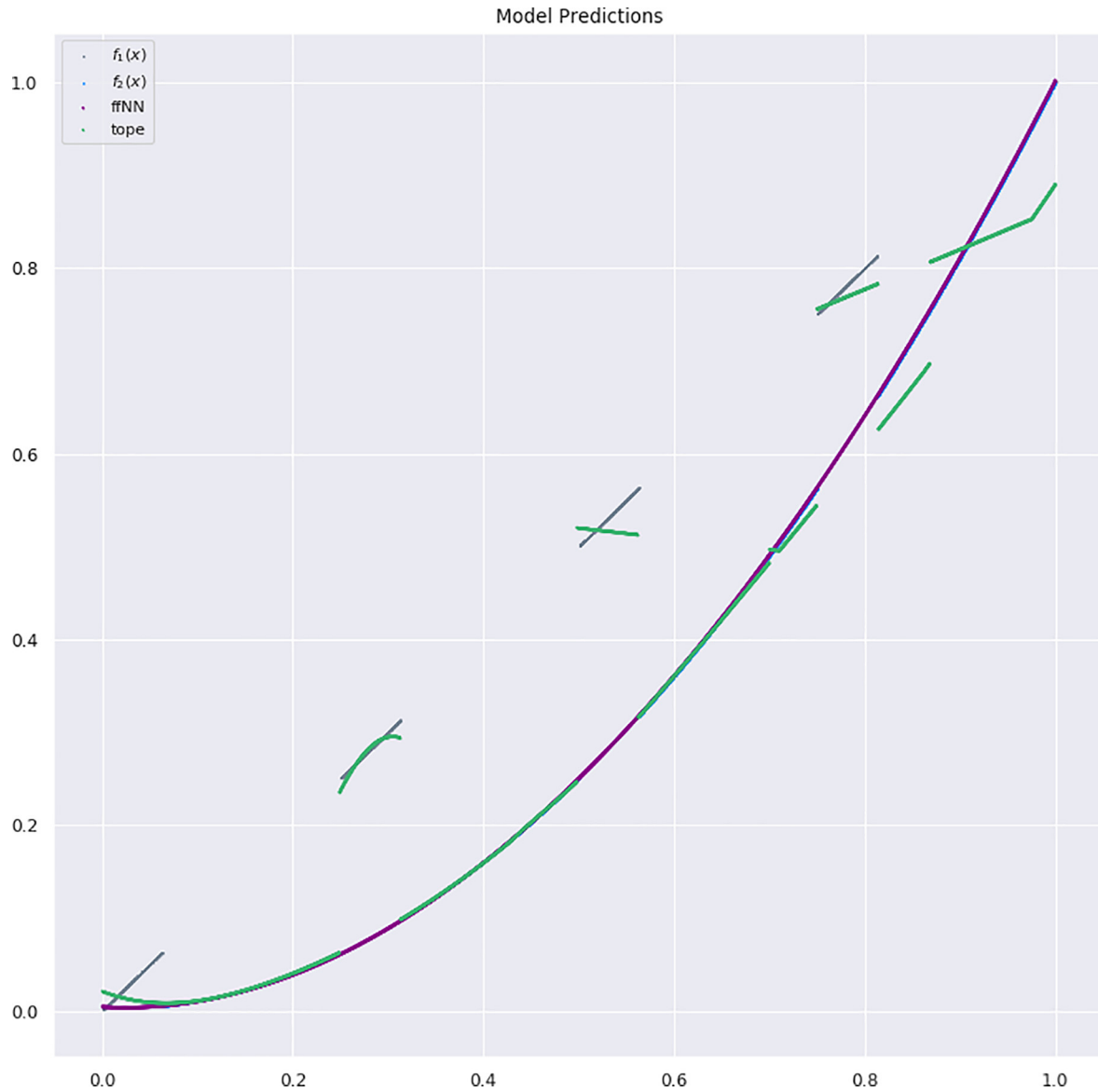


Fig. 1. Approximation of sub-patterns by FFNN.

An instance of the PCNN architecture is illustrated in Fig. 2. The illustration depicts an architecture from a  $d = 3$  dimensional input space to a  $D = 4$  dimensional output space, with 2 “deep zero-sets”  $(\hat{K}_n)_{n=1}^2$ , and, accordingly, two sub-patterns  $\hat{f}_1$  and  $\hat{f}_2$  defining the respective sub-pattern on their respective deep zero-sets. The deep zero-sets are built by feeding the deep classifier  $\hat{c}$  into the discontinuous unit  $x \mapsto x \cdot I_{(\gamma,1]} \circ \sigma_{\text{sigmoid}}$ . Fig. 2 highlights that each trainable part of the network, namely the sub-patterns  $\hat{f}_1$  and  $\hat{f}_2$  and the deep classifier  $\hat{c}$ , all process any input data *independantly* and therefore can be parallelized. The outputs  $\hat{y}^{(1)} \triangleq \hat{f}_1(x_1, x_2, x_3)$ ,  $\hat{y}^{(2)} \triangleq \hat{f}_2(x_1, x_2, x_3)$ , and  $\hat{C} \triangleq \hat{c}(x_1, x_2, x_3)$  of each parallelizable sub-patterns then fed into the discontinuous unit:

$$(\hat{y}^{(1)}, \hat{y}^{(2)}, \hat{C}) \mapsto (\hat{y}^{(1)} I_{(\gamma,1]} \circ \sigma_{\text{sigmoid}}(\hat{C}_1) + (\hat{y}^{(2)} I_{(\gamma,1]} \circ \sigma_{\text{sigmoid}}(\hat{C}_2),$$

which simultaneously defines the deep zero-sets and decides which sub-pattern ( $\hat{f}_i$ ) is to be activated. Fig. 2 shows that the sub-patterns  $\hat{f}_1$  definition of PCNN need not have the configuration of hidden

units. Indeed when training a PCNN, described in Section 4, we will see that each of these sub-patterns is trainable in parallel to one another and therefore their widths and depths may be selected independently of one another.

### 2.2. Set-valued analysis

When performing binary classification, we want to learn which  $x \in X$  belongs to a fixed subset  $K \subseteq X$ . Thus, a classifier  $\hat{c} : X \rightarrow \{0, 1\}$  is typically trained to approximate  $K$ 's indicator function  $I_K$ . In this case, the results of [46] guarantee that  $I_K$  can be approximated point-wise with high probability. However, even if the approximation guarantees are strengthened and  $\hat{c}$  (deterministically) approximates  $I_K$  point-wise, the approximation of  $K$  can be wrong.

**Example 2.1.1.** Let  $X = [-1, 1]$ ,  $K = [0, 1]$ , and  $\{K_k\}_{k=1}^\infty$  be the constant sequence of subsets of  $X$  given by  $K_k \triangleq (0, 1)$ . Then, the pointwise limit of  $I_{K_k}$  equals to  $I_{(0,1)}$ . Thus, the points  $\{0, 1\}$  are always misclassified; even asymptotically.

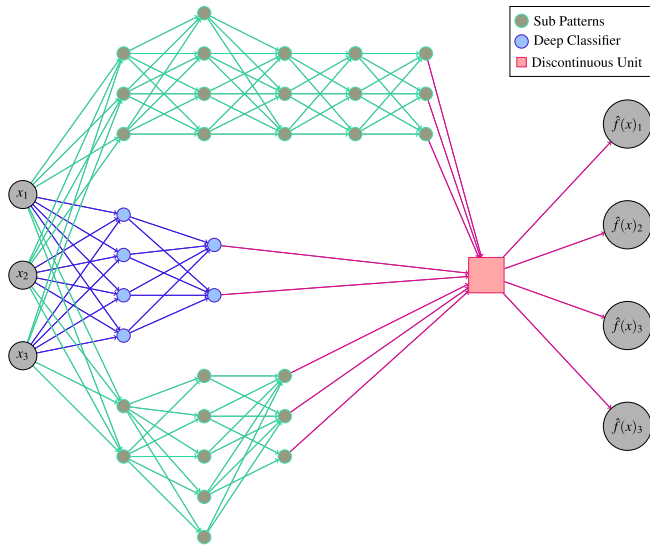


Fig. 2. A PCNN  $\hat{f}$  from  $\mathbb{R}^3$  to  $\mathbb{R}^4$  with two sub-patterns.

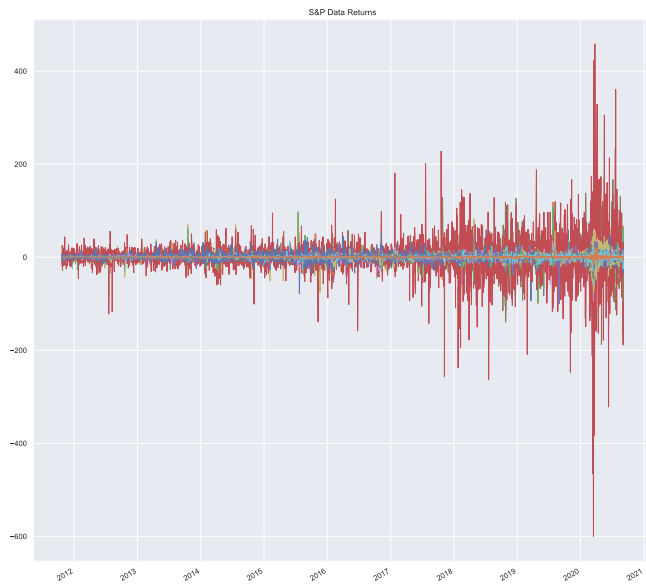


Fig. 3. Four Year Returns of SnP500 Constituents' Returns.

The issue emphasised by Example (2.1.1) is that the points  $\{0, 1\} \subset K$  are limits of some sequence of points in the approximating sets  $K_k$ ; for example, of the respective sequences  $\{\frac{1}{k}\}_{k=1}^\infty$  and  $\{1 - \frac{1}{k}\}_{k=1}^\infty$ . Thus, a correct mode of convergence for sets should rather qualify a limiting set as containing all the limits of all sub-sequences  $\{x_{k_j}\}_{j=1}^\infty$  where  $x_k \in K_k$  for each  $k \in \mathbb{N}^+$ . This is implied, see [47], by the convergence according to the Hausdorff distance  $d_H$  defined for any pair of subsets  $A, B \subseteq X$  via:  $d_{H|X}(A, B) \triangleq \max \{ \sup_{a \in A} \|a - B\|, \sup_{b \in B} \|A - b\| \}$ , where  $\|a - B\| \triangleq \inf_{b \in B} \|a - b\|$  (similarly, we define  $\|A - b\| \triangleq \|b - A\|$ ). The Hausdorff distance between  $A$  and  $B$  represents the largest distance which can be traversed if a fictive adversary assigns a starting point from either of  $A$  or  $B$ . Whenever it is clear what  $X$  is, then we simply denote the Hausdorff distance  $d_{H|X}$  on  $X$  by  $d_H$ . As dis-

cussed in [47], the Hausdorff distance defines a metric on the set  $\text{Comp}(X)$ , whose elements are non-empty compact subsets of  $X$  with Hausdorff distance equipped with the metric  $d_H$ .

### 2.3. Partitions

Fix  $N \in \mathbb{N}^+$  with  $N > 1$ . Throughout this paper, a partition  $\{K_n\}_{n=1}^N$  of the input space  $\mathbb{R}^d$ , means a collection of compact subsets of  $\mathbb{R}^d$ . We use the term "partition" loosely to mean any finite family  $\{K_n\}_{n=1}^N$  of non-empty compact subsets of  $X$ . Therefore, unless otherwise specified, we do not require that  $\text{int}[K_n] \cap \text{int}[K_m]$  to be disjoint. Thus, our results include that situation as a special case.

### 3. Main results

Our main results are now presented. All proofs are relegated to the paper's appendix.

#### 3.1. Universal approximation guarantees

In this section, we examine the asymptotic approximation capabilities of the PCNN model, and we compare and contrast it to that of FFNNs. Two types of approximation theoretic results are considered. The first examines the asymptotic approximation capabilities of  $\text{PCNN}_{d,D}^\sigma$  if the architecture's parameters were simultaneously optimizable. However, as discussed in the paper's introduction, this cannot be done with most available (stochastic) gradient descent-type algorithms. Therefore this class of universal approximation results describes the "gold standard" of what  $\text{PCNN}_{d,D}^\sigma$  could be expected to approximate.

However, since  $\text{PCNN}_{d,D}^\sigma$  are specifically designed to be trainable in a two-step decoupled procedure (via Meta-Algorithm 1 below) to avoid these non-differentiability issues. The next class of universal approximation results are specifically designed to quantify the approximate capabilities of  $\text{PCNN}_{d,D}^\sigma$  when approximating piecewise continuous functions of the form (1). These results are broken into two stages. First, we introduce a space of piecewise continuous functions on which a meaningful universal approximation theorem can be formulated, and we examine some of the properties of these new spaces. Next, we show that  $\text{PCNN}_{d,D}^\sigma$  is universal in this space, and we quantify its approximation efficiency, analogously to [48]. Lastly, we show that, just as in the "gold standard case," FFNNs are not dense in this space; whenever we are approximating piecewise continuous functions with  $N > 1$  in (1).

##### 3.1.1. Gold standard: maximum approximation capabilities

We find that  $\text{PCNN}_{d,D}^\sigma$  can approximate many more functions uniformly than  $\text{NN}_{d,D}^\sigma$ . We do this by comparing the largest space in which  $\text{PCNN}_{d,D}^\sigma$  is universal, to the largest space in which  $\text{NN}_{d,D}^\sigma$  is universal. Since universality (or density) is an entirely topological property, then we compare these two spaces using purely topological criteria. We observe the following.

**Lemma 3.0.1.** Every  $f \in \text{PCNN}_{d,D}^\sigma$  is bounded on  $X$ ; i.e.:  $\sup_{x \in X} \|f(x)\| < \infty$ .

We need an ambient space to make our comparisons. Thus, we begin by viewing  $\text{PCNN}_{d,D}^\sigma$  within the Banach space of all bounded functions from  $X$  to  $\mathbb{R}^D$ , denoted by  $\mathcal{B}(X, \mathbb{R}^D)$ , equipped with the following norm:

$$\|f\|_\infty \triangleq \sup_{x \in X} \|f(x)\|.$$

We use  $\overline{\text{PCNN}_{d,D}^\sigma}$  to denote the closure of  $\text{PCNN}_{d,D}^\sigma$  in  $\mathcal{B}(X, \mathbb{R}^D)$ ; that is,  $\overline{\text{PCNN}_{d,D}^\sigma}$  denotes the collection of all  $f \in \mathcal{B}(X, \mathbb{R}^D)$  for which there is a convergent sequence  $\{f_k\}_{k=1}^\infty \subset \text{PCNN}_{d,D}^\sigma$  with limit  $f$ . In other words,  $\overline{\text{PCNN}_{d,D}^\sigma}$  and  $\overline{\text{NN}_{d,D}^\sigma}$  are the largest space of bounded functions on  $X$  in which  $\text{PCNN}_{d,D}^\sigma$  and  $\text{NN}_{d,D}^\sigma$  are respectively universal, with respect to  $\|\cdot\|_\infty$ .

Next, we show that  $\overline{\text{NN}_{d,D}^\sigma}$  is relatively small in comparison to  $\overline{\text{PCNN}_{d,D}^\sigma}$ . To make this comparison, we appeal to an opposite concept to universality (density), i.e.: *nowhere denseness*, which means that the only open subset of  $\overline{\text{PCNN}_{d,D}^\sigma}$  contained within  $\overline{\text{NN}_{d,D}^\sigma}$  is the empty-set.

**Example 3.0.2.** The set of linear models  $\{ax + b : a, b \in \mathbb{R}\} \subset C(\mathbb{R})$  is nowhere dense in  $C(\mathbb{R})$ . In contrast, the polynomial models  $\{\sum_{n=0}^N a_n x^n : a_n \in \mathbb{R}, N \in \mathbb{N}\}$  are dense in  $C(\mathbb{R})$  (see [49]).

Nowhere dense sets are topologically negligible. Consequently,  $\overline{\text{NN}_{d,D}^\sigma}$  is topologically negligible in  $\overline{\text{PCNN}_{d,D}^\sigma}$ .

**Theorem 3.0.1.** [The PCNN is Asymptotically More Expressive] Let  $\sigma \in C(\mathbb{R})$  be non-polynomial, and  $X = [0, 1]^d$ . Then,  $\overline{\text{NN}_{d,D}^\sigma}$  is nowhere dense in  $\overline{\text{PCNN}_{d,D}^\sigma}$ .

The next result shows that  $\overline{\text{PCNN}_{d,D}^\sigma}$  is large in the sense that it is not separable, this means that any dense subset thereof cannot be countable. Examples of separable Banach spaces arising in classical universal approximation results include  $C(\mathbb{R}^d, \mathbb{R}^D)$  studied in [10,32], the space of Lebesgue  $p$ -integrable functions  $L^p(\mathbb{R}^d)$  on  $\mathbb{R}^d$  studied in [50], or the Sobolev spaces on bounded domains in  $\mathbb{R}^d$  as in [51]. The most well-known example of a non-separable space is  $L^\infty(\mathbb{R}^d)$ , in which [52] have shown that  $\overline{\text{NN}_{d,D}^\sigma}$  is not dense whenever  $\sigma$  is continuous.

**Theorem 3.0.2** (PCNN s are Highly Expressive, Asymptotically). Let  $\sigma \in C(\mathbb{R})$  be non-polynomial, and  $X = [0, 1]^d$ . Then the following hold:

- (i)  $\overline{\text{PCNN}_{d,D}^\sigma}$  is not separable,
- (ii)  $C(X, \mathbb{R}^D) \subset \overline{\text{PCNN}_{d,D}^\sigma}$ .

**Remark 3.0.1.** Theorems 3.0.1 and 3.0.2 formulate universal approximation theorems that express the maximum approximation capabilities of the PCNN relative to  $\overline{\text{NN}_{d,D}^\sigma}$ .

We complete this portion of our discussion here by observing that even simple piecewise continuous functions (with more than one piece) cannot be uniformly approximated by FFNNs with a continuous activation function.

**Proposition 3.0.1** (Deep Feedforward Networks are Not Universal in  $\mathcal{B}(X, \mathbb{R}^D)$ ). If  $\sigma \in C(\mathbb{R})$  and  $K_1 = [0, \frac{1}{2}], K_2 \triangleq [\frac{1}{2}, 1]$  are given. Then, For every  $\epsilon > 0$  there exists an  $f_\epsilon$  of the form (1) satisfying:

$$\inf_{f \in \overline{\text{NN}_{d,D}^\sigma}} \sup_{x \in [0,1]} \|f(x) - f_\epsilon(x)\| \geq \epsilon.$$

Though the space  $\mathcal{B}(X, \mathbb{R}^D)$  does provide a concrete environment for comparing the maximum approximation capabilities of different deep neural models, it is nevertheless ill-suited to the approximation of piecewise continuous functions. These are the two following reasons. This is because approximation of any  $f \in \mathcal{B}(X, \mathbb{R}^D)$  can not be decoupled, in the sense that any piecewise function of the form (1) can not be approximated by approximating the  $f_1, \dots, f_N$  and the  $K_1, \dots, K_N$  in separate steps. Rather uniform approximation necessitates that both must be approximated in the same step, but we cannot do this due to the discontinuous structure of our model. Furthermore, the uniform norm does not genuinely reflect any of the “sub-pattern pattern” structure of (1), and rather it only views  $f$  as a typical bounded function. Accordingly, we now introduce a space of piecewise continuous functions and a mode of convergence for piecewise functions capable of detecting when piecewise functions have differing numbers of “sub-patterns” (defined below) and whose mode of convergence is amenable to a two-step optimization procedure in which trains the  $f_n$  are the  $K_n$  are separately.

### 3.2. PCNNs are universal approximators of piecewise continuous functions

We introduce our space of piecewise continuous functions as well the notion of convergence of piecewise continuous functions. Then, we show that the PCNNs are universal for this mode of convergence. PCNNs are shown to optimize a certain concrete upper bound to the “distance function” of piecewise continuous functions; this upper bound reflects our proposed algorithm (described in the next section). Lastly, we show that just like in the space  $\mathcal{B}(X, \mathbb{R}^D)$ , FFNNs with continuous activation functions are not universal in the space of piecewise continuous functions.

#### 3.2.1. The space of piecewise continuous functions

The motivation for our mathematical framework for describing and approximating piecewise continuous functions can be motivated by the fact that many FFNNs implement the same continuous function [28,53]. For instance, [26,54,53] distinguishes between FFNN’s (abstract) representation, which is an  $N + 1$ -tuple  $\{W_n\}_{n=1}^{N+1}$  of composable affine functions  $W_n$ ; where these function’s parameters encode the information required to implement an FFNN once the activation function is specified. Indeed, upon fixing  $\sigma$ , the realization of an (abstract) FFNN  $\{W_n\}_{n=1}^{N+1}$  is the  $\hat{f} \in C(\mathbb{R}^d, \mathbb{R}^D)$  defined by:

$$\hat{f}(x) = W_{N+1} \circ \sigma \dots \circ \sigma \bullet W_1(x). \tag{3}$$

The important point here is that, once  $\sigma$  is fixed, every (abstract) FFNN representation induces a unique FFNN realization via (3). However, the converse is generally false. For instance, in [53], it is shown that if  $\sigma$  is the ReLU activation function,  $x \mapsto \max\{0, x\}$ , then any FFNN realization has infinitely many FFNN representations since ReLU layers can implement the identity map on  $\mathbb{R}^D$ . Analogously to FFNNs, every piecewise continuous function of the form (1) has infinitely many different representations in terms of parts (defined below). Therefore, we introduce our theory of piecewise continuous functions in analogy with the above discussion, and we emphasize the distinction between a piecewise continuous function’s representation and its realization.

*The Space of Piecewise Continuous Functions.* In a direct analogy, we define an (abstract) representation of a piecewise continuous function with  $N$  parts to be  $((f_n, K_n)_{n=1}^N)$  where  $f_n \in C(X, \mathbb{R}^D)$  and  $K_n \in \text{Comp}(X)$ ; for  $n = 1, \dots, N$ . The set of all abstract representations of piecewise continuous functions is therefore  $\bigcup_{N \in \mathbb{N}_+} [C(X, \mathbb{R}^D) \times \text{Comp}(X)]^N$ . We define the realization of any piecewise continuous function’s representation  $((f_n, K_n)_{n=1}^N)$  to be the following element  $R((f_n, K_n)_{n=1}^N)$  of  $\mathcal{B}(X, \mathbb{R}^D)$  defined by:

$$R((f_n, K_n)_{n=1}^N)(x) \triangleq \sum_{n=1}^N f_n(x) I_{K_n}(x). \tag{4}$$

The relationship (4) between a piecewise function and its representation(s) is typically non-trivial. As the next example shows, if  $N > 1$  then there may be infinitely many different representations of the same piecewise function.

**Example 3.0.3.** [Non-Uniqueness of Representation] Let  $f(x) = x^2 I_{[0,1]} + e^x I_{[1,2]}$  and  $N = 4$ . For each  $0 < r < 1$ ,  $((x^2, [0, \frac{1}{r}]), (x^2, [\frac{1}{r}, 1]), (-\frac{1}{r^2}, \{\frac{1}{r}\}), (e^x, [1, 2])), ((x^2, [0, 1]), (e^x, [1, 2])),$  and  $((e^x, [1, 2]), (x^2, [0, 1]))$  all represent  $f$ .

We would like to quantify the distance between any two piecewise continuous functions which is invariant to the choice of representation. We would equally like our “distance function” to be capable of detecting if and when two piecewise continuous functions require a different minimal number of parts to be represented. In this way, our metric should be capable of separating any two functions  $f$  and  $g$  if  $f$  is fundamentally more discontinuous than  $g$  is. This is made rigorous through the following; note, the infimum of  $\emptyset$  is defined to be  $\infty$ .

**Definition 3.1.** [Minimal Representation Number] For any  $f \in \mathcal{B}(X, \mathbb{R}^D)$ , its *minimal representation number* is:

$$N(f) \triangleq \inf \left\{ N : \exists ((f_n, K_n)_{n=1}^N) \in \bigcup_{N \in \mathbb{N}_+} [C(X, \mathbb{R}^D) \times \text{Comp}(X)]^N : f = R((f_n, K_n)_{n=1}^N) \right\}.$$

Many functions  $f \in \mathcal{B}(X, \mathbb{R}^D)$  can be arbitrarily complicated; these are beyond the scope of our analysis and can not be uniformly approximated. Here, infinite complexity is quantified by  $N(f) = \infty$ .

**Example 3.1.1.** Let  $f = \sum_{n \in \mathbb{N}_+} n I_{[(n+1)^{-1}, n^{-1}]}$ ; then  $N(f) = \infty$ .

We consider functions as in Example 3.1.1, pathological and, this is because such functions do not admit any representation of the form (1). The following result guarantees that if a piecewise function  $f$  has a representation of the form (1), then it must admit at least one such minimal representation.

**Proposition 3.1.1.** If  $f \in \mathcal{B}(X, \mathbb{R}^D)$  admits a representation of the form (1) then  $N(f) \in \mathbb{N}_+$  exists.

**Example 3.1.2.** If  $f \in C([0, 1], \mathbb{R})$  and  $g = I_{[0, \frac{1}{2}]} + I_{[\frac{1}{2}, 1]}$  then  $N(f) = 1$  and  $N(g) = 2$ .

Example 3.1.2 implies that if  $N(f) > 1$  then  $f$  is discontinuous; however,  $N(f) = 1$  does not imply continuity.

**Example 3.1.3.** Let  $f = I_{[0, \frac{1}{2}]} \in \mathcal{B}([0, 1], \mathbb{R})$ , then we note that  $N(f) = 1$ .

Next, we construct a “distance function”, on the set of piecewise functions on  $X$  with finite minimal representation number. Our construction is reminiscent quotient of metric spaces (see [55]) where different functions are identified as being part of the same “equivalence class”. We first equip  $\bigcup_{N \in \mathbb{N}_+} [C(X, \mathbb{R}^D) \times \text{Comp}(X)]^N$  with the following  $[0, \infty]$ -valued function:

$$d_{\text{Step1}}((f_n, K_n)_{n=1}^N, (f_{n'}, K_{n'})_{n=1}^{N'}) \triangleq \begin{cases} \max_{1 \leq n \leq N} \max \{ \|f_n - f_{n'}\|_\infty, d_H(K_n, K_{n'}) \} & : N = N' \\ \infty & : N \neq N' \end{cases} \tag{5}$$

Next, we use the map  $d_{\text{Step1}}$  to construct a “distance function” on  $\mathcal{B}(X, \mathbb{R}^D)$  which expresses the similarity of any two function  $f, g \in \mathcal{B}(X, \mathbb{R}^D)$  therein in terms of their most similar and efficient yet compatibility representations. A representation  $F := (f_n, K_n)_{n=1}^N \in \bigcup_{N \in \mathbb{N}_+} [C(X, \mathbb{R}^D) \times \text{Comp}(X)]^N$  is understood as being efficient if  $\#F \triangleq N \approx N(f)$ .

Likewise, two representations  $F$  and  $G$ , respectively representing  $f$  and  $g$ , are thought of as being compatible if  $\#F = \#G$  (thus,  $d_{\text{Step1}}(F, G) < \infty$ ) and similar if  $d_{\text{Step1}}(F, G) \approx 0$ . In this way, we may quantify the “distance” between any two bounded functions in  $\mathcal{B}(X, \mathbb{R}^D)$  by searching through all compatible representations, which simultaneously penalizes any such representation for its inefficiency. We define this “distance function”, or divergence, as follows.

**Definition 3.2.** [Piecewise Divergence] The *piecewise divergence* between any two  $f, g \in \mathcal{B}(X, \mathbb{R}^D)$  is defined as:

$$D_{PC}(f|g) \triangleq \inf_{R(F)=f, R(G)=g} \underbrace{d_{\text{Step1}}(F, G)}_{\text{RepresentationSimilarity}} + \underbrace{|\#F - N(f)| + |\#G - N(g)|}_{\text{CostofComparingInefficientRepresentations}} \tag{6}$$

We call  $D_{PC}$  a divergence since even if is symmetric, unlike a metric, it need not satisfy the triangle inequality. Divergences form a common method of quantifying distance in machine learning. Notable examples given by Bregman divergences (see [56] for applications in regularization, [57] for applications in clustering, and [58] for applications in Bayesian estimation). Examples of a divergences are the Kullback–Leibler divergence of [59] and, more generally,  $f$ -divergences which quantify the information shared between different probabilistic quantities [60,61]. Similarly, the divergence  $D_{PC}$  allows us to define the space of piecewise continuous functions by identify all “finitely complicated bounded functions” with their “subpatterns” and the “parts” on which they are defined.

Let  $\mathcal{B}_+(X, \mathbb{R}^D)$  consist of all  $f \in \mathcal{B}(X, \mathbb{R}^D)$  with  $N(f) < \infty$ . We formalize our “space of piecewise continuous functions”, denoted by  $PC(X, \mathbb{R}^D)$ , to be the set of equivalence classes of  $f \in \mathcal{B}_+(X, \mathbb{R}^D)$  where any two functions  $f, g \in \mathcal{B}_+(X, \mathbb{R}^D)$  are identified if  $D_{PC}(f, g) = 0$  and  $D_{PC}(g, f) = 0$ .

Unlike  $\mathcal{B}(X, \mathbb{R}^D)$ , the space  $PC(X, \mathbb{R}^D)$  is well-suited to our approximation problem. This is because, for any  $f \in PC(X, \mathbb{R}^D)$ , the function  $D_{PC}(f|\cdot)$  is upper-bounded by the error of approximating each  $f_n$  and each  $K_n$  individually (using some optimally efficient  $((f_n, K_n)_{n=1}^{N(f)})$  approximate representation of  $f$ ). Most notably, unlike the uniform metric on  $\mathcal{B}(X, \mathbb{R}^D)$ , the approximation error in approximating each  $f_n$  and each  $K_n$  is fully decoupled from one another in this upper-bound of  $D_{PC}(f|\cdot)$ . Thus, we may approximate each  $f_n$  and then each  $K_n$  in two separate steps.

**Proposition 3.2.1** (Decoupled Upper-Bound on  $D_{PC}(f|\cdot)$ ). Fix  $f \in PC(X, \mathbb{R}^D)$  and suppose that

$((f_n, K_n)_{n=1}^{N(f)}) \in [C(X, \mathbb{R}^D) \times \text{Comp}(X)]^{N(f)}$  represents  $f$ . For any  $((\hat{f}_n, \hat{K}_n)_{n=1}^{N(f)}) \in [C(X, \mathbb{R}^D) \times \text{Comp}(X)]^{N(f)}$  the following holds:

$$D_{PC}\left(f \mid \sum_{n=1}^{N(f)} \hat{f}_n I_{\hat{K}_n}\right) \leq \sum_{n=1}^{N(f)} \|f_n - \hat{f}_n\|_\infty + \sum_{n=1}^{N(f)} d_H(K_n, \hat{K}_n). \quad (7)$$

The decoupled upper-bound for the piecewise divergence  $D_{PC}$  derived in Proposition 3.2.1 will now be used to show that PCNNs are universal approximators of piecewise continuous functions.

As a final point of interest and motivation for our main result, we demonstrate that sequences of FFNNs in  $\mathbf{NN}_{d,D}^\sigma$  generally do not converge, with respect to the piecewise divergence, to piecewise functions with at-least two parts. This is because the PC divergence between any two functions  $f$  and  $g$  is lower-bounded by  $|N(f) - N(g)|$ . In particular, since every feed-forward network is in  $C(X, \mathbb{R}^D)$ , then the result follows from Example 3.1.2.

**Proposition 3.2.2.** [FFNNs are Not Universal Piecewise Continuous Functions] Fix  $\sigma \in C(\mathbb{R})$ . For each  $f \in PC(X, \mathbb{R}^D)$ , if  $N(f) > 1$  then:

$$1 \leq \inf_{f \in \mathbf{NN}_{d,D}^\sigma} D_{PC}(f|\hat{f}). \quad \text{Furthermore, if } X = [0, 1] \text{ then, } f = I_{[0,1]} + I_{[\frac{1}{3}, \frac{2}{3}]} \in PC(X, \mathbb{R}) \text{ and } N(f) > 1.$$

### 3.2.2. Approximating piecewise continuous functions by PCNNs

Our results, we require the following regularity condition on  $\sigma \in C(\mathbb{R})$  introduced in [32].

**Assumption 3.2.1.**  $\sigma \in C(\mathbb{R})$  is not affine. There is an  $x_0 \in \mathbb{R}$  at which  $\sigma$  is continuously-differentiable and  $\sigma'(x_0) \neq 0$ .

We may now state our main result, which shows that PCNNs is a universal approximator of piecewise continuous function with respect to the piecewise continuous divergence,  $D_{PC}$ . In particular, PCNNs is strictly more expressive than FFNNs since Proposition 3.2.2 proved that FFNNs are not universal with respect to  $D_{PC}$ .

**Theorem 3.2.1** (PCNNs are Universal Approximators of Piecewise Continuous Functions). Fix  $\sigma$  satisfying Assumption 3.2.1. Let  $f \in PC([0, 1]^d, \mathbb{R}^D)$ ,  $0 < \gamma < 1$ , and  $0 < \epsilon$ . There exist a  $\hat{f} \in PC(X, \mathbb{R}^D)$

with representation  $\hat{F} \triangleq ((\hat{f}_n, \hat{K}_n)_{n=1}^N)$  where each  $\hat{f}_n \in \mathbf{NN}_{d,D}^\sigma$  of width at-most  $d + D + 2$ ,  $\hat{K}_n \triangleq \{x \in X : I_{[\gamma, 1]} \circ \sigma_{\text{sigmoid}} \circ \hat{c}(x)_n\}$  and where  $\hat{c} \in \mathbf{NN}_{d, N(f)}^\sigma$  has width at-most  $d + N(f) + 2$ , such that:

$$D_{PC}(f|\hat{F}) < \epsilon. \quad (8)$$

A key step towards establishing Theorem 3.2.1, which we now highlight, is deriving the following fact that our deep zero-sets are universal in  $\text{Comp}(X)$  with respect to the Hausdorff metric  $d_H$ , thereon. This result is the first result describing the universal approximation of compact sets defined by a deep neural model.

**Theorem 3.2.2** (Deep Zero-Sets are Universal Compact Sets). Let  $\sigma$  satisfy Assumption 3.2.1,  $\emptyset \neq K_1, \dots, K_N \subseteq X$  be compact,  $0 < \epsilon$ , and set  $\gamma \triangleq \sigma_{\text{sigmoid}}^{-1}(2^{-1}\epsilon)$ . There is an  $\hat{c} \in \mathbf{NN}_{d, N}^\sigma$  of width at-most  $d + N + 2$  for which the deep zero-sets  $\hat{K}_n \triangleq \{x \in X : I_{[\gamma, 1]} \circ \sigma_{\text{sigmoid}} \circ \hat{c}(x)_n\}$  simultaneously satisfy:

$$\max_{n=1, \dots, N} d_H(K_n, \hat{K}_n) \leq \epsilon. \quad (9)$$

We introduce our parallelizable procedure for training the PCNNs and we investigate its theoretical properties.

### 3.3. The training meta-algorithm

Let  $\mathbb{X}$  be a non-empty set of training data in  $\mathbb{R}^d$  and let  $L : \mathbb{R}^d \rightarrow [0, \infty)$  quantify the learning problem

$$\inf_{\hat{f} \in \text{PCNN}_{\mathbb{X} \times \mathbb{X}}} L(\hat{f}(x)). \quad (10)$$

Even if  $L$  is smooth, we cannot use gradient-descent type methods to optimize (10) (in-sample on  $\mathbb{X}$ ) since the map  $I_{[\gamma, 1]}$  makes each  $\hat{K}_n$  into a discontinuous function, and by extension  $\hat{f}$  need not be differentiable (even in the generalized sense of [62]). Meta-Algorithm 1 proposes an approach for training the PCNN which avoids passing gradient updates through the indicator function  $I_{[\gamma, 1]}$  in  $\hat{K}_n$  by decoupling the training of the  $\{\hat{f}_n\}_{n=1}^N$  and the  $\{\hat{K}_n\}_{n=1}^N$ .

---

#### Algorithm 1 Train PCNN.

---

- 1: **Input:** Training data  $\mathbb{X}$ , Number of Parts  $N$ , Depth parameters  $J_1, J_2, W \in \mathbb{N}_+$ , FFNN training subroutine GET\_FFNN, GET\_PARTITION subroutine.
  - 2: **Output:** Trained PCNN  $f^\star \triangleq \sum_{n \leq N} f_n^\star(\cdot) \cdot I_{K_n^\star}(\cdot)$
  - 3:  $\{\mathbb{X}_n\}_{n=1}^N \leftarrow \text{GET\_PARTITION}(\mathbb{X})$
  - 4: **for**  $n \leq N$  **do**
  - 5:  $f_n^\star \leftarrow \text{GET\_FFNN}(\mathbb{X}_n, L, J, W)$
  - 6: **end for**
  - 7: **for**  $x \in \mathbb{X}$  **do**
  - 8:  $l_{n,x} \triangleq I(L(f_n^\star(x)) \leq \min_{m \leq N} L(f_m^\star(x)))$
  - 9: **end for**
  - 10:  $c^\star \leftarrow \text{GET\_FFNN}(\mathbb{X}, H(\cdot|l_n, \cdot), J_2, W)$
  - 11: **for**  $n \leq N$  **do**
  - 12: Define deep zero-sets  $K_n \triangleq \{x \in \mathbb{R}^n : c_n^\star(x) \leq 2^{-1}\}$
  - 13: **end for**
- 

Step 1 (line 3) initializes partitions of the training data, according to an extraneously given subroutine GET\_PARTITION, an example of which we propose in Subroutine 2 below. Step 2 (lines 4-6) optimizes the networks  $\hat{f}_n$ .

Step 4 (lines 7-9) identifies which optimized sub-pattern  $\hat{f}_n$  best performs on any given input and adjusts the partitions. Step 5 (li-



nes 10–12) interprets the  $\{l_{n,x}\}_{n,x}$  in  $\{0, 1\}$  as labels and trains the  $\hat{c}_n$  defining the  $\hat{K}_n$  (see Definition 2.1) as a classifier predicting when  $f_n^{\hat{c}}$  offers the best cross-entropy:  $H(y|l_{n,x}) \triangleq l_{n,x} \ln(y)(1 - l_{n,x}) \ln(1 - y)$ , amongst the  $\{f_n^{\hat{c}}\}_{n=1}^N$  (justified by Theorem 3.2.3 below).

Though there are many possible clustering algorithms which can stand-in for the subroutine GET\_PARTITION in Meta-Algorithm 1, we present a novel geometric option with desirable properties.

### 3.3.1. Initializing and training the deep zero-sets

Suppose that we can operate under the “geometric priors” that nearby points tend to belong to the same part  $K_n$  and, given  $N$ , the parts  $\{K_n\}_{n=1}^N$  are as efficient as possible. This means that  $\{K_n\}_{n=1}^N$  should partition  $X$  while having the smallest possible boundaries. In particular, if  $\mathbb{X}$  is representative of  $\{K_n\}_{n=1}^N$ , then GET\_PARTITION seeks to partition the  $\mathbb{X}$  into  $N$  parts  $\{\mathbb{X}_n\}_{n=1}^N$  while minimizing the distance between every pair of data-points in each  $\mathbb{X}_n$ .

This problem is known as the min-cut problem and it is a well-studied problem in computer science. In particular, the authors of [63] show that this is an NP-hard problem. Nevertheless, exploiting the max-flow min-cut duality (see [64]) a randomized-polynomial time algorithm which approximately solves the min-cut problem with high probability is developed in [65]. Furthermore, the algorithm tends to assign nearby points to the same part, where the probability of this happening depends linearly on the distance between those two points.

One stand-in for GET\_PARTITION, which we describe in Subroutine 2, modifies the procedure of [65] to fit our setting. Key properties of our variant are described in Proposition 3.2.3. In particular, Proposition 3.2.3 (v) states that with high-probability, two nearby data-points in  $\mathbb{X}$  are mapped to the same sub-pattern.

We use  $\Delta(\mathbb{X}) \triangleq \min_{x,y \in \mathbb{X}; x \neq y} \frac{1}{2} \|x - y\|$  to denote the minimum distance between any pair of training data-points and  $\bar{\Delta}(\mathbb{X})$  the mean distance between distinct training data, given by  $\bar{\Delta}(\mathbb{X}) \triangleq \frac{1}{\#\{(x,y) \in \mathbb{X}^2: x \neq y\}} \sum_{(x,y) \in \mathbb{X}^2: x \neq y} \|x - y\|$ . In Algorithm 2, the randomness arises from  $\alpha$  which contributes to the random radius in Line 7 that forms the data partitions. The minimum portion of the data required in each partition is denoted by  $q$ . Line 3, shuffles the training data. Line 7 use the random radius defined by  $\alpha$  to form the partitions of the data and line 8 extends the partition of the data to forms parts of the input space. Lines 10–14 ensure each part is not too large relative to the others.

---

#### Algorithm 2 GET\_PARTITION.

---

```

1: Input: Training data  $\mathbb{X}, q \in (0, 1)$ 
2: Sample  $\alpha$  uniformly from  $(\frac{1}{2}, \frac{1}{4})$ 
3: Pick a bijection  $\pi : \{1, \dots, \#\mathbb{X}\} \rightarrow \mathbb{X}$ 
4: Compute  $\Delta(\mathbb{X})$  and  $\bar{\Delta}(\mathbb{X})$ 
5:  $\mathbb{X}' \leftarrow \mathbb{X}$ 
6: loop
7:  $\mathbb{X}_n \triangleq \{z \in \mathbb{X}' : \|z - \mathbb{X}'_0\| < \alpha \bar{\Delta}(\mathbb{X})\}$ 
8:  $K_n^{\mathbb{X}} \triangleq \{z \in \mathbb{R}^d : (\exists x \in \mathbb{X}_n) \|x - z\| \leq \Delta(\mathbb{X})\}$ 
9:  $\mathbb{X}' \leftarrow \mathbb{X}' - \mathbb{X}_n$ 
10: if  $\frac{\#\mathbb{X}'}{\#\mathbb{X}} \leq q$ 
11:    $\mathbb{X}_n \triangleq \mathbb{X}'$ 
12:    $K_n^{\mathbb{X}} \triangleq \{z \in \mathbb{R}^d : (\exists x \in \mathbb{X}_n) \|x - z\| \leq \Delta(\mathbb{X})\}$ 
13:   BREAK

```

```

14: end if
15: end loop
16: Output:  $\{\mathbb{X}_n\} \triangleq \{\mathbb{X}_n \neq \emptyset\}, N \triangleq \#\{\mathbb{X}_n\},$ 
 $\{K_n^{\mathbb{X}}\} \triangleq \{K_n^{\mathbb{X}} \neq \emptyset\}$ 

```

---

**Proposition 3.2.3** (Properties of Subroutine 2). Let  $\mathbb{X}, q = 1$ , and  $\{K_n^{\mathbb{X}}\}_{n=1}^N$  be as in Subroutine 2, and fix  $\hat{f}_1, \dots, \hat{f}_N \in \mathbb{NN}_{d,D}^{\sigma}$ . Set  $f = \sum_{n=1}^N f_n I_{K_n^{\mathbb{X}}}$ . The following hold:

- (i) For  $x_1, x_2 \in \mathbb{X}$ ,  $\mathbb{P}(\min_{n=1, \dots, N} \max_{i=1, 2} \|\hat{f}_i(x_i) - \hat{f}_n(x_i)\| = 0) \geq 1 - \frac{8(\ln(\#\mathbb{X})+1)}{\Delta(\mathbb{X})} \|x_1 - x_2\|$ ,
- (ii) Subroutine 2 terminates in polynomial time,
- (iii) For  $n \leq N$ ,  $\text{int}[K_n^{\mathbb{X}}] \neq \emptyset$ ,
- (iv) If  $n \neq m$  then  $K_n^{\mathbb{X}} \cap K_m^{\mathbb{X}} = \emptyset$

Our final result guarantees that, given trained models  $\{f_n^{\hat{c}}\}$ , there exists  $\{K_n^{\hat{c}}\} \subset \text{Comp}(X)$  which optimizes the PCNN's performance with arbitrarily high-probability. We quantify this by a fixed Borel probability measure  $\mathbb{P}$  on  $X$ .

**Theorem 3.2.3.** [Existence: Performance Optimizing Partition] Fix  $\{\hat{f}_n\}_{n=1}^N \in \mathbb{NN}_{d,D}^{\sigma}$  and  $L \in C(\mathbb{R}^D, [0, \infty))$  for which  $L(0) = 0$ . There exists a compact subset  $X_{\delta, \mathbb{P}} \subseteq X$  and a partition of  $X_{\delta, \mathbb{P}}$  satisfying:

- (i)  $\mathbb{P}(X_{\delta, \mathbb{P}}) \geq 1 - \delta$ ,
- (ii) For  $n \leq N$  and every  $x \in K_n^{\hat{c}}$ ,  $L(\hat{f}_n(x)) = \min_{m \leq N} L(\hat{f}_m(x))$ .

Moreover, if  $\mathbb{X} \subseteq X_{\delta, \mathbb{P}}$  and  $L(\hat{f}_n(x)) < \min_{n \neq \bar{n}, \bar{n} \leq N} L(\hat{f}_{\bar{n}}(x))$  for each  $n \leq N, x \in \mathbb{X}_n$  then,  $\mathbb{X}_n \subseteq K_n^{\hat{c}}$ .

**Remark 3.2.1** (Approximation of Optimal Partition). Applying Theorem 3.2.1, we conclude that for every  $\epsilon > 0$  there is a PCNN  $\hat{f}$  such that  $D_{PC}(\sum_{n=1}^N \hat{f}_n I_{K_n^{\hat{c}}} | \hat{f}) < \epsilon$ ; where the  $\hat{f}_1, \dots, \hat{f}_N$  are from Meta-Algorithm 1 (4–6).

## 4. Numerical experiments

We evaluate the PCNN's performance on three different regression tasks. The goal of our experiments is twofold. The first objective is to show that the PCNN trained with Meta-Algorithm 1 better approximates the  $\{K_n\}_{n=1}^N$  defining the piecewise function compared to the benchmark models. The second goal is to show that the model offers a predictive advantage when the function being approximated is discontinuous.

### 4.1. Implementation details

In the following, We implement PCNN, trained with Algorithm (1) and subroutine 2, against various benchmarks. The first benchmark is the FFNNs, which we use to evaluate the predictive performance improvement obtained by turning to PCNNs and utilizing a discontinuous layer. The second class of benchmarks focuses on the effectiveness of the PCNN's structure itself. We consider two

naive alternatives to the proposed model design. Since PCNN can be viewed as an ensemble model, we compare its predictive performance against a bagged model (FFNN-BAG) wherein the user trains  $N$  distinct feed-forward networks  $\hat{f}_1, \dots, \hat{f}_N$  on the deep zero-sets generated using subroutine 1 which are summed together to construct the bagged model  $\hat{f}_{\text{FFNNBAG}}(x) \triangleq \sum_{n=1}^N \hat{f}_n(x)$ . This benchmark has the benefit of distinguishing parts of the inputs space via the  $\{K_n\}_{n=1}^N$ , instead of naively grouping them into one estimator. Next, to evaluate the effectiveness of the deep partitions, we consider the (FFNN-LGT) model, which is identical to the PCNN in structure except that the deep classifier  $\sigma$  defining the deep partitions  $\hat{K}_n$  is replaced by a simple logistic classifier.

The model quality is evaluated according to their test predictions and the learned model's complexity/parsimony. Prediction quality is quantified by mean absolute error (MAE), mean squared error (MSE), and mean absolute percentage error (MAPE), each evaluated on the test set. The models are trained by using the mean absolute error (MAE). The model *complexity* is assessed by the total number of parameters in each model (#Par), the average number of parameters processing each input (#Par/x) and the training times, either with parallelization (P. time) or without (L. time).

Our numerical experiments emphasize the scope and compatibility of Meta-Algorithm1 with most FFNN training procedures. Fittingly, the experiments in Fig. 1 and Sections 4.3 take GET\_FFNN to be the ADAM stochastic optimization algorithm of [14] whose behaviour is well-studied [66]. The experiments in Sections 4.2 and 4.4 train the involved FFNNs by randomizing all their hidden layers and only training their final “linear readout” layer. This latter method is also well understood [67–71].

In the latter experiments, we also benchmark PCNNs against a deep feedforward network with randomly generated hidden weights and linear readout trained with ridge-regression (FFNN-RND). The FFNN-RND and FFNN benchmarks put the speed/precision trade-off derived from randomization into perspective. This allows us to gauge the PCNN architecture's expressiveness as it is still the most accurate method even after this near-complete randomization; in comparison, the FFNN-RND method's predictive power will reduce when compared to the FFNN trained with ADAM.

## 4.2. Learning discontinuous target functions

It is well known that the returns of most commonly traded financial assets forms a discontinuous trajectory [23,72,73]; the origin of these “jump discontinuities” are typically abrupt “regime switches” of the underlying market dynamics caused by news or other economic and behavioural factors. The magnitude of discontinuous behaviour depends on the idiosyncrasies of the particular financial assets. The next experiments illustrate that PCNNs can effectively model discontinuous function, with varying degrees of discontinuities, by illustrating that they can replicate the returns of assets with different levels of volatility; i.e.: the degree to which any asset's returns fluctuate.

### 4.2.1. Mild discontinuities: SnP500 market index replication

In this experiment we train the PCNN, along with the benchmarks, to predict the next-day SnP500 market index's returns using the returns of all its constituents; i.e. the stocks of the 500 largest companies publically traded in the NYSE, NASDAQ, Cboe BZX exchanges. The dataset consists of 2 years of daily closing data ending on September 9<sup>th</sup> 2020. The test set consists of the final two weeks. All returns are computed with the daily closing prices.

Since the principle parameter, extending FFNNs to PCNNs, is the integer  $N$ , controlling the number of parts and sub-patterns

thereon. We study the effect of varying the number of parts when training a PCNN using Algorithms 1 and 2. Figs. 4(c) explore the effect of varying  $N$  on the considered performance metrics. We find that the PCNN achieves the best performance amongst the considered models while relying on the smallest number of trainable parameters. Thus, PCNN is the most efficient model. Throughout this ablation experiment, the PCNNs deployed in Figs. 4(c) are forced to have a comparable number of active neurons, with a fixed minimum width to ensure expressibility. This is necessary, for example, even in the case when there is a single-part whereon PCNNs essentially coincide with FFNNs [74,75].

Figs. 4(a) and 4(b) show that, once enough parts have been built into the PCNN, it outperforms the feedforward models. From Fig. 4 (b), we also see that the parallelizability and randomization of Subroutine 2's 3<sup>rd</sup> step enables a relatively small increase in parallelized training times compared to the FFNN, even when  $N$  increases. Furthermore, since each  $\hat{f}_n$  are progressively narrowed as  $N$  increases, then the training time further accelerates as the  $\hat{f}_n$  are built using progressively fewer neurons.

Fig. 4(b) shows that, P. time increases as the number of parts defining PCNN do, this is because training  $\hat{c}$  and Subroutine 2 is not parallelizable and scale in  $N$ ; albeit not dramatically. Thus, the number of parts defining the PCNNs' predictive power, as expressed through the MAE and MSE losses, rises before tapering off. This is because a higher number of parts allows more regions of discontinuities to be captured, but since our experiment fixed the total number of neurons defining the PCNNs then each sub-pattern can become too narrow to support expressivity as the number of parts becomes large. see Table 1. see Fig. 5.

Table 2 shows that the models learning the partitions of the input space, i.e. FFNN-LGT and PCNN, enhance the predictive performance. Furthermore, the flexibility offered by the paradigm of Meta-Algorithm1 further improves the prediction of the next-day Bitcoin closing price.

We examine the effect of the number of partitions; we repeat the experiment with a fixed number of neurons distributed amongst the subpatterns  $\{\hat{f}_n\}_{n=1}^N$  with the test set consisting of the final two weeks of February 20<sup>th</sup> 2018. We also force the network  $\hat{c}$  in (2), defining all the deep zero-sets  $\{\hat{K}_n\}_{n=1}^N$ , to scale at a rate of  $N^{-1}$  to ensure an even comparison between the FFNNs (i.e.: PCNN with a single part) and the genuine PCNNs with multiple parts. see Fig. 7.

From Fig. 6(a), we see that PCNNs has a lower test-set MAE than PCNNs with fewer parts, and in particular they outperform PCNNs with 1 part; i.e.: feedforward neural networks. Figs. 6(a) and 6(c) show that the PCNNs require fewer neurons to produce their predictions. Moreover, Table 2 shows that any given input  $x \in X$  is processed by far fewer neurons than are available in the entire PCNN; thus any input is first “triaged” by the network  $\hat{c}$  then assigned to its correct deep zero-set and processes by the correct subpattern of the PCNN specialized to that part of the input space.

### 4.3. Beating an expert partition

Often one has access to a partition of the training inputs derived from expert insight. This experiment evaluates the impact of using Subroutine 2 for GET\_PARTITION in Meta-Algorithm1 in comparison to using the expert partition. The experiment will be on the Kaggle housing dataset [76] where there is a commonly accepted expert partition [77]. see Fig. 8.

We compare the PCNN model also trained using Meta-Algorithm1 and the ADAM optimizer [14] for GET\_FFNN but now with the expert partition in place of Subroutine 2. This benchmark

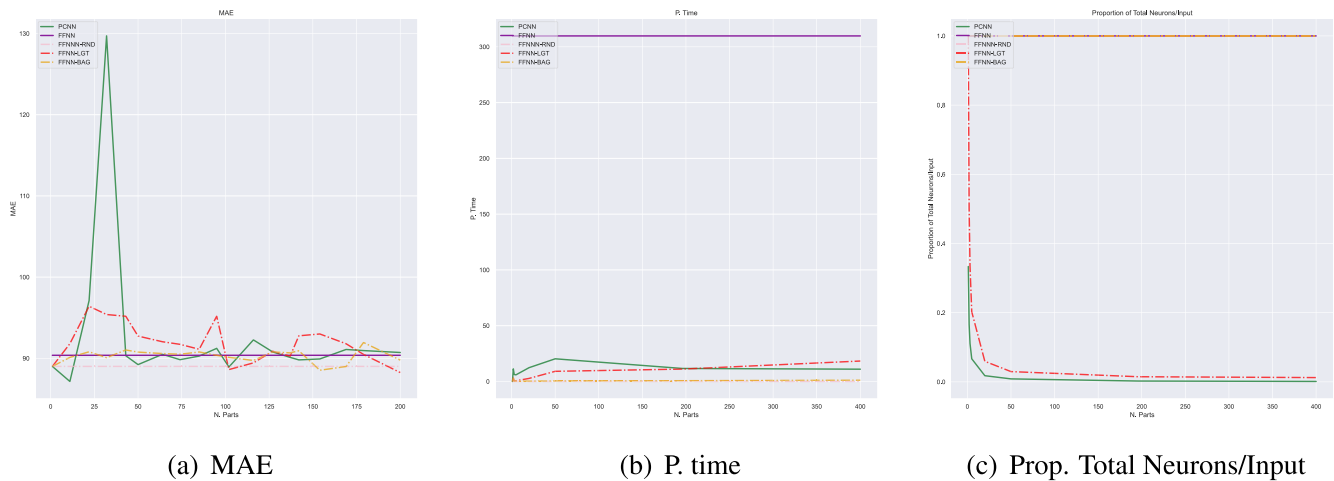


Fig. 4. Performance as Function of Number of Parts with Variable Number of Neurons.

Table 1 Predictive Performance and Complexity Metrics.

	MAE	P. Time	L. Time	#Par/x	#Par
FFNN	9.0e+2	–	8.0e+2	6.0e+3	6.0e+3
FFNN-RND	1.3e+4	–	8.7e-3	6.0e+3	6.0e+3
FFNN-BAG	9.0e+2	1e-1	2.4e-1	5.6e+6	5.6e+6
FFNN-LGT	9.1e+2	2.5e+1	2.7e-1	6.1e+6	5.5e+6
PCNN	8.7e+2	9.1e+1	9.2e+1	5.1e+4	5.4 e+5

(PCNN + EXP) quantifies how well the partition initialization of Subroutine 2 and the partition updating of Meta-Algorithm 1 (steps 10–12) performs against the commonly accepted partition of that dataset. The PCNN + EXP benchmark takes the [77] partition and then trains an FFNN independently on each part using [14] before recombining them with the discontinuous unit (Fig. 2 in red), see Fig. 9.

Table 2 Predictive Performance and Complexity Metrics.

	MAE	P. Time	L. Time	#Par/x
FFNN	9.6e+2	–	1.2e+1	1.2e+4
FFNN-RND	7.6e+4	–	2.7e-3	1.2e+4
FFNN-BAG	4.5e+3	3.1e-2	3.9e-1	4.87 + 6
FFNN-LGT	1.0e+3	1.4e-1	5.0e-1	2.0e+6
PCNN	8.2e+2	1.0e+1	1.4e+1	1.2e+5

The impact of the partition updating of Meta-Algorithm 1 (steps 10–12), given a good initialization of the partition, is quantified by another benchmark (PCNN + EXP + UP). The PCNN + EXP + UP benchmark is trained just as PCNN + EXP, but it also incorporates the partition updating in Meta-Algorithm 1 before regrouping the trained FFNNs by the discontinuous unit (Fig. 2 in red). Thus, PCNN + EXP + UP measures the impact of Meta-Algorithm 1’s partition updating steps. Since the PCNN + EXP’s parts are not implemented by neurons, as are the other benchmark models, reporting #Par/x in this experiment would not be accurate; rather, Table 6 reports the total number of parameters (#Par) in each model.

The PCNN + EXP + UP benchmark contrasts against the FFNN-LGT benchmark, also considered in the above experiments, which quantifies the impact of our partitioning procedure against a naively chosen one with no good prior initialization. The benchmarks of the previous experiments are included in Table 6 to gauge the PCNN’s performance.

The PCNN trained with Meta-Algorithm 1 out-predicts all the considered benchmarks; in particular, we note the predictive gap of 4k\$MAE from the second-best benchmark (PCNN + EXP + UP) and a gap of 8k\$from the FFNN benchmark. Thus, no prior knowledge of the input is needed for a successful PCNN deployment using Meta-Algorithm 1.

When examining the importance of proper partitioning, the gap between the FFNN-LGT and the PCNN models shows that a poorly chosen partition still impacts the model’s performance. In more granularity, the PCNN + EXP has an MAE of 3.175e+4 while the PCNN + EXP + UP has a mildly lower MAE of 3.171e+4. This shows that an accurate choice of the subroutine GET.PARTITION is much more impactful in Meta-Algorithm 1 than the updating steps (10–12). The reason for this is that the training of PCNN’s sub-patterns depends on that initialization. Therefore, a poor choice of an initial partition translates to a reduction of the PCNN’s inductive bias. This is validated by the small gap between the FFNN-LGT, the PCNN + EXP, and PCNN + EXP + UP models.

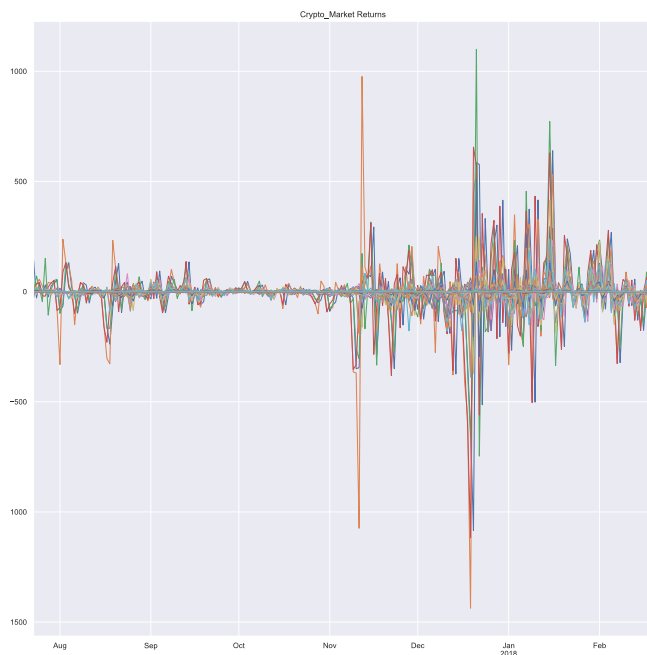


Fig. 5. Two Year Cryptocurrency Returns.

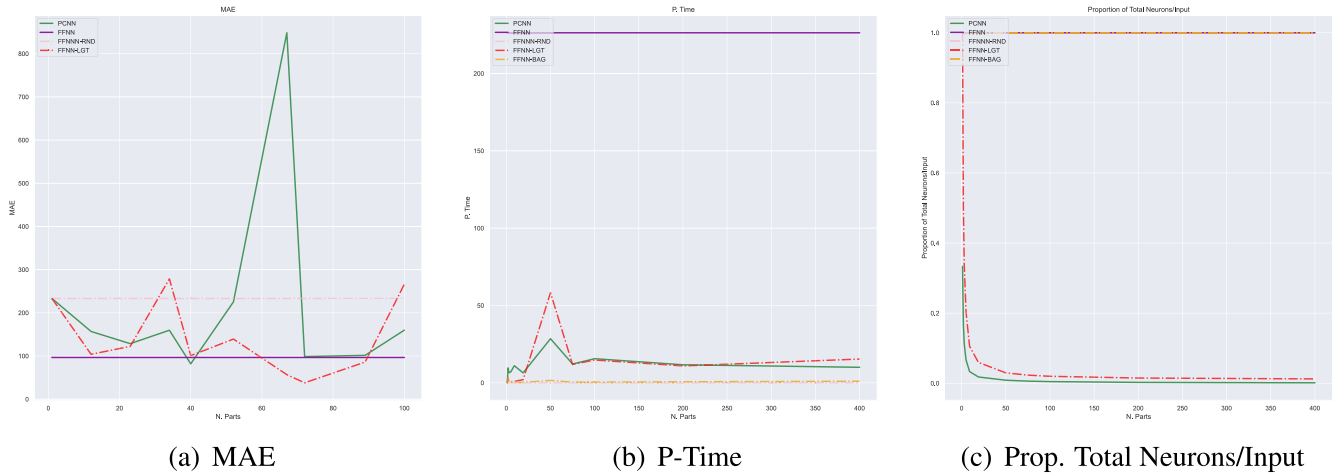


Fig. 6. Performance as Function of Number of Parts with Variable Number of Neurons.

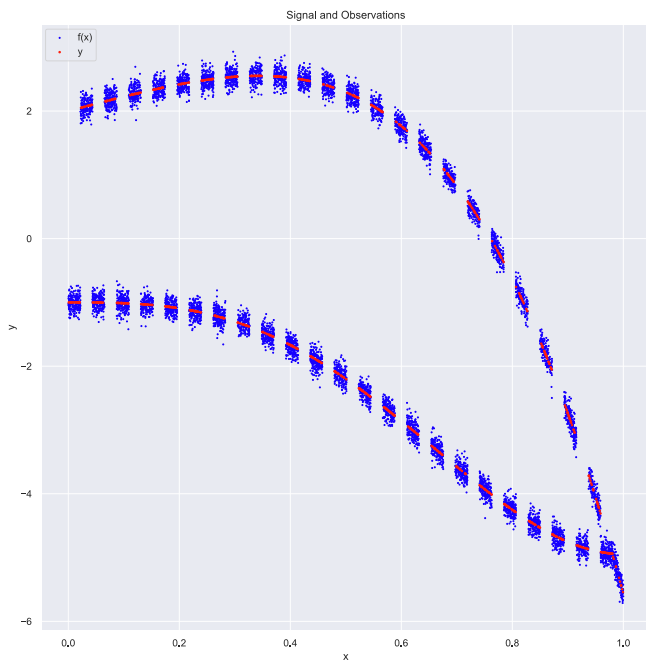


Fig. 7. Test and Noisy Training Data (if  $d = 1 = A$ ).

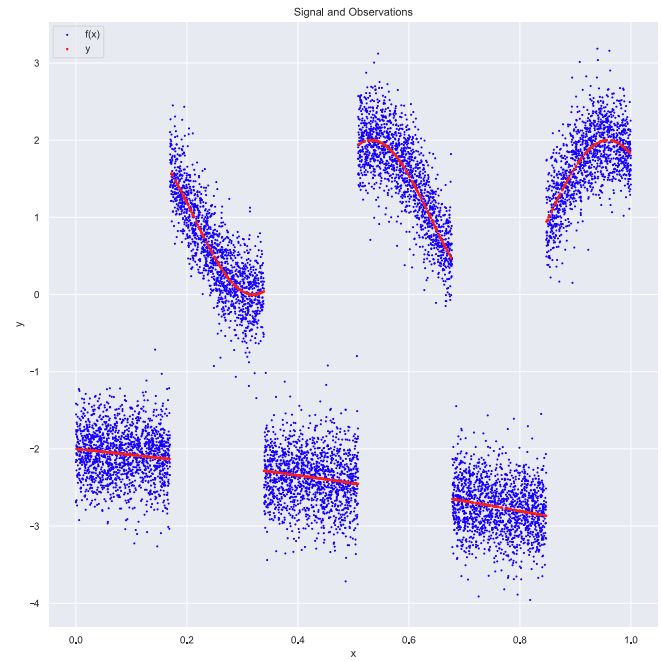


Fig. 8. Test and Noisy Training Data (if  $d = 1 = A$ ).

#### 4.4. Ablation within A controlled environment

We begin by ablating the performance of the PCNN architecture on various synthetic experiments, wherein we may examine the effect of each component of the synthetic data on the proposed model. We study the performance of various learning models when faced with the non-linear regression problem

$$y_n = f(x_n) + \sigma \epsilon_n^v, \tag{11}$$

where for each  $n = 1, \dots, N$ ,  $x_n$  is sampled uniformly from  $[0, 1]^d$ ,  $\sigma \geq 0$  is the variance parameter, and are  $\epsilon_n^v$  are i.i.d. random variables with t-distribution with  $v$  degrees of freedom. We vary the behaviour of  $f$ , the dimensionality  $d$ , the level of noise  $\sigma$ , and the size of  $\epsilon_n^v$ 's extreme values captured by the heavy-tailedness parameter  $v$ , to understand how the PCNN architecture trained according to Subroutine 2 behaves. We consider piecewise continuous  $f$  of the form:

$$f(x) = f_1(Ax)I_{[0,2^{-1}r)}(Ax \bmod r) + f_2(Ax)I_{[2^{-1}r,r)}((Ax \bmod r)),$$

where  $r > 0$  captures the rate at which the sub-patterns  $f_1$  and  $f_2$  interchange and  $A$  is a randomly generated  $D \times 1$ -matrix with i.i.d. standard Gaussian entries. For instance, if  $d = 100$  then there is on average, 10 discontinuities in each direction of the input space. The difficulties in the non-linear regression problem (11) arise from the many discontinuities of  $f$ , the opposing and oscillating trends of the  $f_i$ , the problem's dimensionality, and the heavy-tailedness of its noise.

The previous experiment's predictive results were based on concrete dollar values; however, the outcomes of these experiments are just numerical values. Therefore to maintain interpretability, all performance metrics will be reported as a fraction over the principle benchmark, i.e. the FFNN model's performance metric.

Each experiment also reports the mean total number of neurons processing each of the network's inputs ( $\#Par/x$ ), the total number

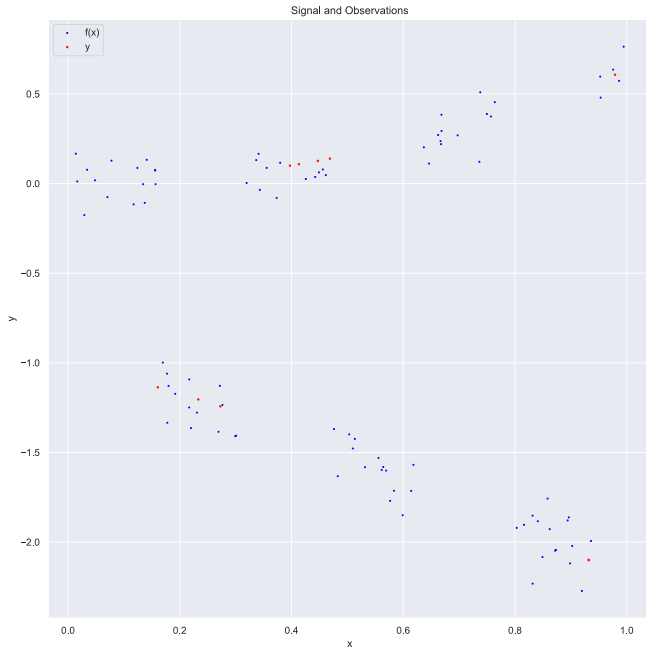


Fig. 9. Test and Noisy Training Data (if  $d = 1 = A$ ).

of parts used in building each neural model, and the quantities  $d$ ,  $\sigma$ , the number of data points,  $v$ , and  $r$ . To frame the irregularity of each function being learned, each table will be accompanied by a plot of the samples from the noiseless target function  $f$  (in red) and the noisy training data (in blue) in the “visualizable case” where  $d = 1$  and  $A = 1$ .

Table 3 Performance Metrics/FFNN - Varying “Discontinuity Rate” ( $r$ ).

	MAE	P. Time	L. Time	#Par/x	#Parts	d	$\sigma$	#Data	$v$	$r$
FFNN	1.00e+00	1.00e+00	–	1.00e+00	1	100	0.01	10000	30	0.25
FFNN-RND	1.00e+04	1.45e-04	–	2.49e-03	1	100	0.01	10000	30	0.25
FFNN-BAG	2.96e+00	1.05e-02	1.77	9.95e-01	400	100	0.01	10000	30	0.25
FFNN-LGT	1.91e+00	1.11e-01	1.87	1.00e+00	400	100	0.01	10000	30	0.25
PCNN	9.97e-01	1.93e-01	1.95	1.99e+00	400	100	0.01	10000	30	0.25
FFNN	1.00e+00	1.00e+00	–	1.00e+00	1	10e+2	0.01	10e+5	30	0.1
FFNN-RND	9.06e+03	5.35e-05	–	2.49e-03	1	10e+2	0.01	10e+5	30	0.1
FFNN-BAG	5.72e+00	3.46e-03	1.58e-1	3.66e-01	147	10e+2	0.01	10e+5	30	0.1
FFNN-LGT	5.98e+00	4.30e-02	1.98e-1	3.67e-01	147	10e+2	0.01	10e+5	30	0.1
PCNN	9.87e-01	7.70e-02	2.31e-1	7.31e-01	147	10e+2	0.01	10e+5	30	0.1

Table 4 Performance Metrics/FFNN – Varying Signal-to-Noise Ratio ( $\sigma, v$ ).

	MAE	P. Time	L. Time	#Par/x	#Parts	d	$\sigma$	#Data	$v$	$r$
FFNN	1.00e+00	1.00e+00	–	1.00e+00	1	10e+2	0.01	10e+5	15	0.25
FFNN-RND	1.69e+01	5.53e-05	–	2.49e-03	1	10e+2	0.01	10e+5	15	0.25
FFNN-BAG	1.18e+00	3.48e-02	21.74	3.70e+00	1400	10e+2	0.01	10e+5	15	0.25
FFNN-LGT	1.11e+00	3.82e-01	22.09	3.72e+00	1400	10e+2	0.01	10e+5	15	0.25
PCNN	8.86e-01	1.33e-01	21.84	7.40e+00	1400	10e+2	0.01	10e+5	15	0.25
FFNN	1.00e+00	1.00e+00	–	1.00e+00	1	10e+2	0.1	10e+5	5	0.25
FFNN-RND	1.70e+01	7.14e-05	–	2.49e-03	1	10e+2	0.1	10e+5	5	0.25
FFNN-BAG	1.21e+00	3.19e-02	27.73	4.98e+00	2000	10e+2	0.1	10e+5	5	0.25
FFNN-LGT	1.19e+00	2.14e-01	27.91	5.00e+00	2000	10e+2	0.1	10e+5	5	0.25
PCNN	8.48e-01	8.92e-02	27.78	9.95e+00	2000	10e+2	0.1	10e+5	5	0.25

#### 4.4.1. Parsing oscillations from discontinues

This experiment examines how well PCNNs can learn discontinuities in the presence of distinct oscillating subpatterns. For this, we set  $f_1(u) = 1 + e^u \cos(u)$  and  $f_2(u) = -1 - u^2 \cos(u)$  and vary  $r$ .

As  $r$  approaches 0,  $f$  contains more regions of discontinuities. Table 4 validates our hypothesis by showing that for small  $r$  the feedforward networks (FFNN and FFNN-RND) have trouble capturing these discontinuities. In contrast, even with a mostly random training procedure and subpatterns generated by relatively narrow layers, the PCNN is expressive enough to bypass these issues.

#### 4.4.2. Learning from noisy data

Next, we examine the PCNN’s predictive performance when there is variable levels of noise. In this experiment, we move between the low and high “signal-to-noise ration” regimes by progressively increasing the variance parameter  $\sigma$  and reducing the parameter  $v$ , which increases the size and frequency of extreme values of the  $e^x$  [78]. In this experiment, we consider an oscillatory pattern  $f_1(u) = 1 + \sin(10u)$  which is difficult to parse from noisy data due to its oscillations and a stable pattern  $f_2(u) = -2 - u^2$ .

Table 3 shows that the PCNN model is capable of producing reliable results even when approximating complicated functions in the presence of a high signal-to-noise ratio. We find that the subpatterns  $\hat{f}_n$  selected when noise is high tend to be narrow and the number of parts tends to be larger. Heuristically, this means that the number of parts selected for PCNN tends to be large as the signal-to-noise ratio lowers and visa-versa as the signal-to-noise ratio increases.

#### 4.4.3. Learning from few training samples

We examine the impact of small sample size on the PCNN model’s performance. In this experiment, we fix a relatively simple discontinuous function  $f_1(u) = x$  and  $f_2(u) = x^2$ , and vary the size of the training dataset (#Data). Fig. 5 shows that, just as with the

**Table 5**  
Performance Metrics/FFNN - Varying #Train. Data. (#Data).

	MAE	P. Time	L. Time	#Par/x	#Parts	d	$\sigma$	#Data	$v$	$r$
FFNN	1.00e+00	1.00e+00	–	1.00e+00	1	1	0.01	10e+2	30	0.25
FFNN-RND	1.28e+00	4.80e-04	–	4.98e-03	1	1	0.01	10e+2	30	0.25
FFNN-BAG	8.81e-01	2.39e-04	2.38e-4	4.98e-03	1	1	0.01	10e+2	30	0.25
FFNN-LGT	8.81e-01	1.02e-04	1.02e-4	4.99e-03	1	1	0.01	10e+2	30	0.25
PCNN	8.81e-01	4.80e-04	4.80e-04	9.95e-03	1	1	0.01	10e+2	30	0.25
FFNN	1.00e+00	1.00e+00	–	1.00e+00	1	1	0.01	1000	30	0.25
FFNN-RND	1.19e+06	1.40e-04	–	6.23e-04	1	1	0.01	1000	30	0.25
FFNN-BAG	3.54e+11	1.64e-02	4.02	4.99e-01	1	1	0.01	1000	30	0.25
FFNN-LGT	1.28e+00	0.01	4.01	5.01e-01	800	1	0.01	1000	30	0.25
PCNN	1.02e+00	1.09e-01	4.11	9.98e-01	800	1	0.01	1000	30	0.25

**Table 6**  
Comparison of PCNN with Partitioning and Prediction Benchmarks.

	Partition Benchmarks				Prediction Benchmarks				
	MAE	L. time	P. time	#Par	MAE	L. time	P. time	#Par	
PCNN+EXP	3.17e+4	8.45e+4	4.12e+4	8.37e+5	FFNN	3.21e+4	9.28e+4	9.28e+4	3.7e+5
PCNN+EXP + UP	3.17e+4	1.58e+5	6.60e+4	1.36e+5	FFNN-BAG	4.95e+4	6.36e+4	2.89e+4	2.8e+4
PCNN	3.13e+4	1.28e+5	9.28e+4	3.0e+4	FFNN-LGT	3.18e+4	6.37e+4	2.90e+4	2.8e+4

other experiments which all used  $10e + 5$  training instances, when the #Data is reduced the PCNN model’s predictive performance remains comparable to that of FFNN model.

Table 5 shows when few training data points are available then fewer parts are required, because there is enough space for a classical FFNN (i.e.: a PCNN with a single part) to interpolate the data even if  $f$  is discontinuous. Nevertheless, as the size of the training dataset increases the FFNN does not have as much flexibility to meander through the training data as the PCNN model does, as is seen both in Table 5 and all the previous experiments.

We see that the experiments which used a stochastic optimization method for the subroutine GET\_FFNN tended to use much fewer partitions than the experiments which randomized the involved FFNN’s hidden layers and only trained their final layer. This validates the theoretical results derived in Theorems 3.0.2,3.2.1 guaranteeing that the PCNNs are more expressive than their FFNN counterparts.

### 5. Conclusion

We introduced a new deep neural model capable of uniformly approximating a large class of discontinuous functions. We provided theoretical universal approximation guarantees for each of the PCNN’s parts and the deep neural models as a whole. We showed that the PCNN’s sub-pattern structure could be exploited to decouple the model’s training procedure, thus, avoided passing gradient updates through the model’s non-differentiable  $I_{(\gamma,1)}$  unit. In this way, the PCNN offers the best of both worlds. It is as expressive as deep neural networks with discontinuous activation functions while simultaneously being trainable with (stochastic) gradient-descent algorithms like feed-forward networks with differentiable activation functions.

### CRedit authorship contribution statement

**Anastasis Kratsios:** Conceptualization, Methodology, Formal analysis, Validation, Investigation, Writing - original draft, Writing - review & editing, Visualization, Supervision, Project administration. **Behnoosh Zamanlooy:** Conceptualization, Methodology, For-

mal analysis, Investigation, Writing - review & editing, Visualization.

### Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Acknowledgement

This work was supported by the ETH Zürich Foundation. Anastasis Kratsios was also supported by the European Research Council (ERC) Starting Grant 852821–SWING.

### Appendix A. Proof of the negative result for FFNNs

The first section in our appendix contains the proofs of results illustrating the limitations of FFNNs in approximating piecewise continuous functions.

**Proof** (Proof of Proposition 3.2.2). By definition (i.e., (6)) we have that  $\inf_{R(G)=g} |N(f) - G| \leq D_{PC}(f|\hat{f})$ . Now, since  $1 < N(g)$ , then

$$|N(f) - N(g)| \leq \inf_{R(G)=g} |N(f) - G| \leq D_{PC}(f|\hat{f}). \tag{A.1}$$

Since  $\sigma \in C(\mathbb{R})$ , then  $\mathbf{NN}_{d,D}^\sigma \subseteq C(X, \mathbb{R}^D)$  and since  $X$  is compact then for each  $\hat{f} \in \mathbf{NN}_{d,D}^\sigma$ ,  $(\hat{f}, I_X)$  is a representation of  $\hat{f}$ . Hence,  $N(f) = 1$  and therefore:

$$1 \leq N(f) - 1 = |N(f) - N(g)|. \tag{A.2}$$

Plugging (A.2) into (A.1) yields the desired lower-bound on  $D_{PC}(f|\hat{f})$  for each  $\hat{f} \in \mathbf{NN}_{d,D}^\sigma$ . Lastly, Example 3.1.2 gives the last claim.  $\square$

Next we turn our attention to the proof of [Proposition 3.0.1](#). [Proposition 3.0.1](#) implied by, and is a special case of, the following general result.

**Assumption 5.0.2** (Regularity of Partition I). The partition  $\{K_n\}_{n=1}^N$  satisfies the following:

- (i) **Non-Triviality:** For each  $n, m \leq N$ ,  $\text{int}(K_n) \neq \emptyset$  and if  $n \neq m$  then  $\text{int}(K_n) \cap \text{int}(K_m) = \emptyset$ .
- (ii) **Pairwise Connectedness:** For every  $n \leq N$  there is some  $\tilde{n} < N$  with  $\tilde{n} \neq n$  and  $K_n \cap K_{\tilde{n}} \neq \emptyset$ ,
- (iii) **Regularity:** For  $n \leq N$  we have  $\overline{\text{int}(K_n)} = K_n$ .

**Proposition 5.0.4.** Let  $\{K_n\}_{n=1}^N$  be a partition of  $X$  satisfying [Assumption 5.0.2](#) and let  $\sigma \in C(\mathbb{R})$ . For every  $\delta > 0$ , there exists some  $f_\delta$  of the form (1) with  $\text{supp}(f_\delta) \subseteq K_1 \cup K_n$ , for some  $1 < n \leq N$ , such that

$$f \in \mathbf{NN}_{d,D}^\sigma \implies \inf_{x \in \bigcup_{k=1}^n K_k} \|f(x) - f_\delta(x)\| \geq \delta.$$

In what follows, for any  $d \in \mathbb{N}_+, x \in \mathbb{R}^d$ , and  $\delta > 0$  we use  $\text{Ball}(x, \delta)$  to denote the set  $\text{Ball}(x, \delta) \triangleq \{z \in \mathbb{R}^d : \|x - z\| < \delta\}$ .

**Proof** (Proof of [Proposition 5.0.4](#)). By [Assumption 5.0.2](#) (ii), there exists some  $1 < n < N$  such that  $K_1 \cap K_n \neq \emptyset$  and by [Assumption 5.0.2](#) (i) we have that  $\text{int}(K_1) \cap \text{int}(K_n) = \emptyset$ . Let  $f_\delta$  be defined by

$$f_\delta = -2c_1 I_{\text{int}(K_1)} + 2c_n I_{\text{int}(K_n)},$$

where  $c_1, c_n \in \mathbb{R}^d$  are such that  $\|c_1 - c_n\| = \delta$ . Thus, for every  $f \in \mathbf{NN}_{d,D}^\sigma$  we compute

$$\begin{aligned} \sup_{x \in K} \|f(x) - f_\delta(x)\| &= \max_{i=1,n} \sup_{x \in \text{int}(K_i)} \|f(x) - f_\delta(x)\| \\ &\geq \max_{i=1,n} \sup_{x \in \text{int}(K_i)} \|f(x) - c_i\|. \end{aligned} \quad (3)$$

Let  $f \in \mathbf{NN}_{d,D}^\sigma$ , then there exist composable affine functions  $W_j, \dots, W_1$  such that  $f = W_j \circ \sigma \bullet \dots \bullet \sigma \bullet W_1$ . Since  $\sigma$  is continuous, each  $W_j$  is continuous, and since the composition of continuous functions is again continuous then  $f$  is continuous. Since  $f$  is continuous, then its supremum on any bounded open subset  $B \subseteq \mathbb{R}^d$  is equal to the maximum of  $f$  over the closure of  $B$  in  $\mathbb{R}^d$ ; hence we refine (3) to

$$\begin{aligned} \sup_{x \in K} \|f(x) - f_\delta(x)\| &= \max_{i=1,n} \sup_{x \in \text{int}(K_i)} \|f(x) - c_i\| \\ &= \max_{i=1,n} \max_{x \in K_i} \|f(x) - c_i\|. \end{aligned} \quad (4)$$

Suppose that  $\sup_{x \in K} \|f(x) - f_\delta(x)\| < \delta$  for some  $f \in \mathbf{NN}_{d,D}^\sigma$ . Then (4) implies that  $f(x) \in B_\delta(c_i)$  for  $i = 1, n$  but by definition of  $c_1$  and  $c_n$  we had that  $\|c_1 - c_n\| = \delta$  and therefore  $B_\delta(c_1) \cap B_\delta(c_n) = \emptyset$ , a contradiction. Hence, there does not exist some  $f \in \mathbf{NN}_{d,D}^\sigma$  satisfying  $\sup_{x \in K} \|f(x) - f_\delta(x)\| < \delta$ .  $\square$

## Appendix B. Proof of supporting results

**Proof** (Proof of [Lemma 3.0.1](#)). If  $f \in \mathbf{PCNN}_{d,D}^\sigma$ , then by definition there exist  $f_1, \dots, f_N \in \mathbf{NN}_{d,D}^\sigma$  and some  $K_1, \dots, K_N \subseteq X$ , such that  $f = \sum_{n=1}^N f_n I_{K_n}$ . Since  $\mathbf{NN}_{d,D}^\sigma \subseteq C(X, \mathbb{R}^D)$ , since  $X$  is compact, and since

every continuous function achieves its maximum on a compact space then, for every  $n = 1, \dots, N$ ,  $\sup_{x \in X} \|f_n(x)\| < \infty$ . Therefore, we compute

$$\begin{aligned} \sup_{x \in X} \left\| \sum_{n=1}^N f_n(x) I_{K_n}(x) \right\| &\leq \sup_{x \in X} \sum_{n=1}^N \|f_n(x) I_{K_n}(x)\| \\ &= \sup_{x \in X} \sum_{n=1}^N I_{K_n}(x) \|f_n(x)\| \\ &\leq \sup_{x \in X} \sum_{n=1}^N 1 \|f_n(x)\| < \infty. \end{aligned}$$

Thus,  $f$  is bounded on  $X$  if  $f \in \mathbf{PCNN}_{d,D}^\sigma$ .  $\square$

**Proof** (Proof of [Proposition 3.1.1](#)). If  $f \in \mathcal{B}(X, \mathbb{R}^D)$  admits a representation of the form  $\sum_{n=1}^{N'} f_n I_{K_n}$  for some  $N' \in \mathbb{N}_+$ , some  $f_1, \dots, f_{N'} \in C(X, \mathbb{R}^D)$  and some  $K_1, \dots, K_{N'} \in \text{Comp}(X)$  then,

$$\begin{aligned} N(f) &= \inf \left\{ N \in \mathbb{N}_+ : \exists ((f_n, K_n))_{n=1}^N \in \bigcup_{k \in \mathbb{N}_+, k \leq N'} [C(X, \mathbb{R}^D) \times \text{Comp}(X)]^k \right. \\ &\quad \left. : f = R\left(\left((f_n, K_n)\right)_{n=1}^N\right) \right\}. \end{aligned}$$

Since the set  $\{1, \dots, N'\}$  is finite, then it admits a minimum and therefore  $N(f) \in \mathbb{N}_+$  exists.  $\square$

**Proof** (Proof of [Proposition 3.2.1](#)). By definition,  $N(f) = \# \left( (\hat{f}_n, \hat{K}_n) \right)_{n=1}^{N(f)}$ . Therefore, (6) is upper-bounded using  $d_{\text{Step1}}$  of (5) via:

$$\begin{aligned} D_{\text{PC}} \left( f | R \left( \left( (\hat{f}_n, \hat{K}_n) \right)_{n=1}^{N(f)} \right) \right) &\leq \left( \inf_{R \left( \left( (\hat{f}_n, \hat{K}_n) \right)_{n=1}^N \right) = f, N=N(f)} d_{\text{Step1}} \left( \left( (\hat{f}_n, \hat{K}_n) \right)_{n=1}^N, \left( (\hat{f}_n, \hat{K}_n) \right)_{n=1}^N \right) \right) \end{aligned} \quad (B.1)$$

$$\begin{aligned} &= \left( \inf_{R \left( \left( (\hat{f}_n, \hat{K}_n) \right)_{n=1}^N \right) = f, N=N(f)} \max_{1 \leq n \leq N} \max \left\{ \|\hat{f}_n - \hat{f}_n\|_\infty, d_H(\hat{K}_n, \hat{K}_n) \right\} \right) \\ &\leq \max_{1 \leq n \leq N(f)} \max \left\{ \|f_n - \hat{f}_n\|_\infty, d_H(K_n, \hat{K}_n) \right\}. \end{aligned} \quad (B.2)$$

Note that, we have use the fact that  $f \in \mathbf{PC}(X, \mathbb{R}^D)$  to conclude that  $N(f) < \infty$  and therefore the infimum in (B.1) is not vacuously  $\infty$  and we have used the fact that  $\# \left( (\hat{f}_n, \hat{K}_n) \right)_{n=1}^{N(f)} = N(f)$  to conclude that each  $d_{\text{Step1}} \left( \left( (\hat{f}_n, \hat{K}_n) \right)_{n=1}^N, \left( (\hat{f}_n, \hat{K}_n) \right)_{n=1}^N \right) < \infty$  in (B.1) is finite. Next, we observe that for each  $N \in \mathbb{N}_+$  and each  $x \in \mathbb{R}^N$  it follows that  $\max_{1 \leq n \leq N} |x_n| \leq \sum_{n=1}^N |x_n|$ ; thus, we upper-bound on the right-hand side of (B.2) as follows:

$$\begin{aligned} D_{\text{PC}} \left( f | R \left( \left( (\hat{f}_n, \hat{K}_n) \right)_{n=1}^{N(f)} \right) \right) &\leq \max_{1 \leq n \leq N(f)} \max \left\{ \|f_n - \hat{f}_n\|_\infty, d_H(K_n, \hat{K}_n) \right\} \\ &\leq \sum_{n=1}^{N(f)} \max \left\{ \|f_n - \hat{f}_n\|_\infty, d_H(K_n, \hat{K}_n) \right\} \\ &\leq \sum_{n=1}^{N(f)} (\|f_n - \hat{f}_n\|_\infty + d_H(K_n, \hat{K}_n)). \end{aligned} \quad (B.3)$$

Thus, (B.3) implies that the estimate (7) holds.  $\square$

## Appendix C. Proofs of main results

This appendix contains proofs of the paper's main results. We draw the reader's attention to the fact that many of the paper's

results are interdependent and, as such, this appendix organizes our results' proofs in their logical order and that this order may differ from the one used in the paper exposition.

C.1. Proofs concerning the closure of the PCNNs in  $\mathcal{B}(X, \mathbb{R}^D)$

**Proof** (Proof and Discussion of Theorem 3.0.1). Define the subset  $\mathcal{V} \subseteq \mathcal{B}(X, \mathbb{R}^D)$  by,  $f \in \mathcal{V}$  if and only if there exist some  $f_1, f_2 \in C(X, \mathbb{R}^D)$  for which

$$f = \left( I_{\left(\frac{1}{2}, 1\right]} \circ \sigma_{\text{sigmoid}} \circ g \right) f_1 + f_2;$$

where  $g \in \ggg$  is the same as in part (i). Let  $f, \tilde{f} \in \mathcal{V}$  and  $k \in \mathbb{R}$ . Then,

$$f + k\tilde{f} = \left( I_{\left(\frac{1}{2}, 1\right]} \circ \sigma_{\text{sigmoid}} \circ g \right) (f_1 + k\tilde{f}_1) + (f_2 + k\tilde{f}_2) \in \mathcal{V};$$

where, mutatis mutandis, we have represented  $\tilde{f}$  in the form (C.1).

Hence,  $\mathcal{V}$  is a vector space containing  $\overline{\text{PCNN}_{d,D}^\sigma}$  as a proper subspace; where the latter claim follows by taking  $f_2 = 0$  in the representation (C.1) and  $f_1 \in \overline{\text{PCNN}_{d,D}^\sigma} = C(X, \mathbb{R}^D)$  arbitrary. In particular, by construc-

$$\begin{aligned} d_\infty \left( \left( I_{\left(\sigma_{\text{sigmoid}}(g \circ \gamma(t_1)), 1\right] \circ \sigma_{\text{sigmoid}} \circ g \right) f, \left( I_{\left(\sigma_{\text{sigmoid}}(g \circ \gamma(t_2)), 1\right] \circ \sigma_{\text{sigmoid}} \circ g \right) f \right) &= \sup_{x \in X} \left\| \left( I_{\left(\sigma_{\text{sigmoid}}(g \circ \gamma(t_1)), 1\right] \circ \sigma_{\text{sigmoid}} \circ g(x)} \right) f(x) - \left( I_{\left(\sigma_{\text{sigmoid}}(g \circ \gamma(t_2)), 1\right] \circ \sigma_{\text{sigmoid}} \circ g(x)} \right) f(x) \right\| \\ &= \sup_{t \in (0,1)} \left\| \left( I_{\left(\sigma_{\text{sigmoid}}(g \circ \gamma(t_1)), 1\right] \circ \sigma_{\text{sigmoid}} \circ g(\gamma(t)) \right) f(x) - \left( I_{\left(\sigma_{\text{sigmoid}}(g \circ \gamma(t_2)), 1\right] \circ \sigma_{\text{sigmoid}} \circ g(\gamma(t)) \right) f(x) \right\| = \left\| \left( I_{\left(\sigma_{\text{sigmoid}}(g \circ \gamma(t_1)), 1\right] \circ \sigma_{\text{sigmoid}} \circ g(\gamma(t_1)) \right) f(\gamma(t_1)) \right. \\ &\quad \left. - \left( I_{\left(\sigma_{\text{sigmoid}}(g \circ \gamma(t_2)), 1\right] \circ \sigma_{\text{sigmoid}} \circ g(\gamma(t_1)) \right) f(\gamma(t_1)) \right\| > 1. \end{aligned}$$

tion,  $\mathcal{V}$  is a subspace of  $\mathcal{B}(X, \mathbb{R}^D)$ . By the Uniform Limit Theorem [Theorem 21.6] [25]  $C(X, \mathbb{R}^D)$  is closed with respect to  $d_\infty$ ; hence,  $C(X, \mathbb{R}^D)$  a closed proper subset of  $\mathcal{V}$ . Hence, it is not dense therein. Applying [Excercise 11.4.3 (f)] [79] we conclude that  $C(X, \mathbb{R}^D)$  is nowhere dense in  $\mathcal{V}$ . It is therefore sufficient to show that  $\mathcal{V} \subseteq \overline{\text{PCNN}_{d,D}^\sigma}$  to conclude that  $C(X, \mathbb{R}^D)$  is nowhere dense

$\overline{\text{PCNN}_{d,D}^\sigma}$ , and therefore,  $\overline{\text{PCNN}_{d,D}^\sigma}$  is nowhere dense in  $\overline{\text{PCNN}_{d,D}^\sigma}$ .

Let  $f \in \mathcal{V}$ , which we represent according to C.1, and let  $\epsilon > 0$ . By the hypothesized density of  $\overline{\text{PCNN}_{d,D}^\sigma}$  in  $C(X, \mathbb{R}^D)$  there exist some

$$\hat{f}_1^\epsilon, \hat{f}_2^\epsilon \in \overline{\text{PCNN}_{d,D}^\sigma} \text{ satisfying}$$

$$\max_{i=1,2} \|f_i - \hat{f}_i^\epsilon\|_\infty < \frac{\epsilon}{2}. \tag{C.2}$$

Therefore, from (C.2) and (C.1) we compute the estimate

$$\begin{aligned} &\left\| f - \left( I_{\left(\frac{1}{2}, 1\right]} \circ \sigma_{\text{sigmoid}} \circ g \right) \hat{f}_1^\epsilon + \hat{f}_2^\epsilon \right\|_\infty \\ &= \left\| \left( I_{\left(\frac{1}{2}, 1\right]} \circ \sigma_{\text{sigmoid}} \circ g \right) (f_1 - \hat{f}_1^\epsilon) + (f_2 - \hat{f}_2^\epsilon) \right\|_\infty \\ &= \left\| \left( I_{\left(\frac{1}{2}, 1\right]} \circ \sigma_{\text{sigmoid}} \circ g \right) f_1 - \hat{f}_1^\epsilon \right\| + \left\| f_2 - \hat{f}_2^\epsilon \right\|_\infty \\ &= \left\| I_{\left(\frac{1}{2}, 1\right]} \circ \sigma_{\text{sigmoid}} \circ g \right\| \left\| f_1 - \hat{f}_1^\epsilon \right\| + \left\| f_2 - \hat{f}_2^\epsilon \right\|_\infty \\ &\leq 1 \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon. \end{aligned}$$

Hence,  $\mathcal{V} \subseteq \overline{\text{PCNN}_{d,D}^\sigma}$ . The conclude the proof.  $\square$

In what follows, we denote  $d_\infty(f, g) \triangleq \|f - g\|$ .

**Proof** (Proof of Theorem 3.0.2). Since  $\overline{\text{PCNN}_{d,D}^\sigma}$  is dense in  $C(X, \mathbb{R}^D)$ , there

exists an  $f \in \overline{\text{PCNN}_{d,D}^\sigma}$  for which  $f_n(x) > 1$  for each  $n = 1, \dots, D$  and every  $x \in X$ . Similarly, since  $\ggg$  is dense in  $C(X, \mathbb{R})$  then it must contain some non-constant  $g$  for which  $g(x) > 0$  for all  $x \in X$ . Since  $g$  is non-constant and  $X$  is path-connected then the intermediate value theorem ([Theorem 24.3] [25]) implies that there exist a curve  $\gamma : (0, 1) \rightarrow X$  such that

$$g \circ \gamma(t_1) < g \circ \gamma(t_2), \tag{3}$$

for each  $0 < t_1 < t_2 < 1$ . Since  $\sigma_{\text{sigmoid}}$  is injective, then (3) implies that

$$0 < \sigma_{\text{sigmoid}}(g \circ \gamma(t_1)) < \sigma_{\text{sigmoid}}(g \circ \gamma(t_2)) < 1. \tag{4}$$

Consider the subset  $\mathcal{Z} \triangleq \left\{ \left( I_{\left(\sigma_{\text{sigmoid}}(g \circ \gamma(t_1)), 1\right] \circ \sigma_{\text{sigmoid}} \circ g \right) f \right\} \subseteq \overline{\text{PCNN}_{d,D}^\sigma}$ .

By (4), for every  $0 < t_1 < t_2 < 1$  we compute:

Therefore,  $\mathcal{Z}$  is an uncountable discrete subspace of  $\overline{\text{PCNN}_{d,D}^\sigma}$ .

Hence, it cannot contain a countable dense subset (since each singleton in  $\mathcal{Z}$  is an open subset of  $\mathcal{Z}$ ). In other words,  $\mathcal{Z}$  is not separable. Since every subset of a separable space it is itself separable, then we conclude that  $\overline{\text{PCNN}_{d,D}^\sigma}$  is not separable. That is, (i) holds.

Next, we show (ii). Lastly, we show that  $C(X, \mathbb{R}^D)$  is a subset of  $\overline{\text{PCNN}_{d,D}^\sigma}$ . Since  $\ggg$  contains the zero-function  $\zeta(x) = 0$ , for all

$x \in X$ , then for every  $f \in \overline{\text{PCNN}_{d,D}^\sigma}$ , the function  $\tilde{f} \triangleq \left( I_{\left(\frac{1}{3}, 1\right]} \circ \sigma_{\text{sigmoid}} \circ \zeta \right) f$  belongs to  $\overline{\text{PCNN}_{d,D}^\sigma}$ . However, by construction

$$\left( I_{\left(\frac{1}{3}, 1\right]} \circ \sigma_{\text{sigmoid}} \circ \zeta(x) \right) = 1,$$

for all  $x \in X$ . Therefore,  $\tilde{f} \left( I_{\left(\frac{1}{3}, 1\right]} \circ \sigma_{\text{sigmoid}} \circ \zeta \right) f = 1f = f$ . Hence,

$\overline{\text{PCNN}_{d,D}^\sigma} \subseteq \overline{\text{PCNN}_{d,D}^\sigma}$ . Since density is transitive, then it follows that

$C(X, \mathbb{R}^d) \subseteq \overline{\text{PCNN}_{d,D}^\sigma}$ . However, any function in  $\mathcal{Z}$  does not belong to  $C(X, \mathbb{R}^D)$  since each function in  $\mathcal{Z}$  is discontinuous. Thus,



$C(X, \mathbb{R}^D)$  is a proper subset of  $\overline{\text{PCNN}}_{d,D}^{\sigma}$ . This concludes the proof.

□

C.2. Proof of Theorem 3.2.2

The proof of our main universal approximation Theorem (i.e., Theorem 3.2.1) relies on approximating the parts of a minimal representation of the target function  $f \in PC(X, \mathbb{R}^D)$  using the decoupled upper-bound of Proposition 3.2.1. The  $f_1, \dots, f_{N(f)}$  in any such minimal representation are approximated using a classical Universal Approximation Theorem, such as [32]. Next, the  $K_1, \dots, K_{N(f)}$  is such a minimal representation are approximated using the following next Lemma (which quantitatively refines Theorem 3.2.2) guarantees that the deep zero sets (defined in (2)) are dense in  $\text{Comp}(X)$  with respect to the Hausdorff metric. The following is a quantitative refinement of Theorem 3.2.2.

**Theorem 5.0.4** (Deep Zero-Sets are Universal: Quantitative Version). *Let  $\sigma$  satisfy Assumption 3.2.1,  $\emptyset \neq K_1, \dots, K_N \subseteq X$  be compact,  $0 < \epsilon$ , and set  $\gamma \triangleq \sigma_{\text{sigmoid}}^{-1}(2^{-1}\epsilon)$ . There is a FFNN  $\hat{c} \in \text{NN}_{d,N}^{\sigma}$  of width at-most  $d + N + 2$  for which the deep zero-sets  $\hat{K}_n \triangleq \{x \in X : I_{(\gamma,1]} \circ \sigma_{\text{sigmoid}} \circ \hat{c}(x)_n\}$*

$$\max_{n=1, \dots, N} d_H(K_n, \hat{K}_n) \leq \epsilon.$$

In particular, if  $X = [0, 1]^d$  then:

- (i) if  $\sigma = \max\{0, x\}$ , then  $\hat{f}$  can be implemented by a FFNN of constant width  $2d + 10$  and depth  $\mathcal{O}((b\epsilon)^{-d+1})$ ,
- (ii) if  $\sigma \in C^\infty(\mathbb{R})$  then  $\hat{f}$  can be implemented by a narrow FFNN with constant width  $2d + 2$  and depth  $\mathcal{O}(b\epsilon^{-2d})$ ,

where in both cases (i) and (ii),  $b > 0$  is a (possibly different) constant which is independent of  $\sigma, \epsilon$ , and of  $K_1, \dots, K_N$ .

**Remark 5.0.2.** [Implications of Lemma 5.0.4 for Classification] Lemma 5.0.4 strengthens the recent deterministic universal classification result in [Theorem 3.11] [36] and the probabilistic classification results of [46]. Moreover, the approximation does not omit sets of Lebesgue measure 0, as do the results of [80].

**Remark 5.0.3.** [Discussion: Approximation Rates of Theorem 5.0.4] always achieve the optimal depth-rates of [48] In [48], it is shown that there are deep ReLU network approximating of constant width  $2d + 10$  and depth

$$\mathcal{O}\left(\left[\left(\tilde{b}\epsilon\right)^{-d+1}\right]^p\right), \tag{5}$$

the achieved the optimal approximate for any  $\frac{1}{p}$ -Hölder function, for  $p \in [1, \infty)$ ; for some constant  $\tilde{b} > 0$ . We make two observations between comparing the rates of (5) and to our rates in Theorem 5.0.4 (i); which describe a network of the same width whose depth grows at the rate:

$$\mathcal{O}\left((b\epsilon)^{-d+1}\right). \tag{6}$$

First, we note that (6) coincides with the optimal rates of (5) in the case where  $p = 1$ ; that is when the target function is Lipschitz. However, the "double exponential" dependence on  $\epsilon$  and the regularity of the target function is always avoided from our problem;

this is because the distance function  $d(\cdot, K)$  can never be less regular than  $p = 1$ . The case where  $\sigma$  is smooth is similar. Here, we notice that the rate (ii) achieves the rate of [37] only when approximating maximally regular target functions; i.e., Lipschitz functions.

**Proof** (Proof of Lemma 5.0.4). Let  $K_1, \dots, K_N \in \text{Comp}(X), N \in \mathbb{N}, 0 < \gamma < 1$ , and let  $0 < \epsilon$ . For each  $n = 1, \dots, N$ , [Lemma 1.2] [81] guarantees that the map  $C : \mathbb{R}^d \ni x \mapsto (d(\cdot, K_n))_{n=1}^N \in \mathbb{R}^N$  is 1-Lipschitz continuous. Therefore, as  $\sigma$  is non-affine and continuously differentiable at at-least one point with non-zero derivative that point, then [Corollary 42] [37] applies. Hence, there exists a FFNN  $\hat{c} \in \overline{\text{NN}}_{d,N}^{\sigma}$  of width at-most  $d + N + 2$  satisfying:

$$\max_{n=1, \dots, N} \max_{x \in X} |d(x, K_n) - \hat{f}(x)_n| \leq \max_{n=1, \dots, N} \max_{x \in X} |C(x)_n - \hat{c}(x)_n| < 2^{-1}\epsilon. \tag{7}$$

Since  $\sigma_{\text{sigmoid}}$  is continuous and monotone increasing then [Theorem 1] [82] implies that it is injective with continuous inverse on its image; thus, we can define  $\gamma \triangleq \sigma^{-1}(2^{-1}\epsilon)$  and we define  $\hat{K}_n \triangleq \{x \in X : I_{(\gamma,1]} \circ \sigma_{\text{sigmoid}} \circ \hat{c}(x)_n - 1 = 0\}$ . Observe that, for each  $n = 1, \dots, N$ , by (7) we have that:

$$\hat{K}_n = \{x \in X : \hat{c}(x)_n < 2^{-1}\epsilon\}. \tag{8}$$

It is enough to show the claim for an arbitrary  $n = 1, \dots, N$ ; thus, without loss of generality we do so for an arbitrary such  $n$ . Let us compute the Hausdorff distance between  $K_n$  and  $\hat{K}_n$ . If  $x \in K_n$ , then by (7) and the definition of  $d(\cdot, K_n)$  we compute:

$$\hat{f}(x)_n \leq |\hat{c}_n(x) - 0| = |\hat{c}_n(x) - d(x, K_n)| \leq 2^{-1}\epsilon. \tag{9}$$

Thus,  $K_n \subseteq \hat{K}_n$ ; hence,  $d(x, \hat{K}_n) = 0$  whenever  $x \in K_n$ . Whence, by the definition of the Hausdorff distance, we have:

$$d_H(K_n, \hat{K}_n) = \max \left\{ \sup_{x \in K_n} d(x, \hat{K}_n), \sup_{x \in \hat{K}_n} d(x, K_n) \right\} = \sup_{x \in \hat{K}_n} d(x, K_n). \tag{10}$$

It remains to bound the right-hand side of (10). Let  $x \in \hat{K}_n$ . By definition of  $\hat{K}_n$  we have that  $\hat{c}(x)_n \leq 2^{-1}\epsilon$ . Coupling this observation with the estimate (7), we find that for every  $x \in \hat{K}_n$ :

$$d(x, K_n) \leq 2^{-1}\epsilon + |\hat{c}(x)_n| \leq 2^{-1}\epsilon + 2^{-1}\epsilon = \epsilon. \tag{11}$$

Combining the estimate of (11) with (10) yields the desired estimate:  $d(K_n, \hat{K}_n) = \sup_{x \in \hat{K}_n} d(x, K_n) \leq \epsilon$ . Lastly, since  $C$  is 1-Lipschitz then if  $X = [0, 1]^d$  then, the estimate in (i) follows from [Theorem 2 (a) and (b)] [48] and the estimate (ii) holds by [Corollary 42] [37]. □

C.3. Proof of main result – Theorem 3.2.1

We may now return to the proof of Theorem 3.2.1.

**Proof** (Proof of Theorem 3.2.1). Let  $f \in PC(X, \mathbb{R}^D)$  and let  $\epsilon > 0$ . By definition,  $N(f) < \infty$  and therefore by Proposition 3.1.1 there exists some  $((f_n, K_n))_{n=1}^{N(f)} \in [C(X, \mathbb{R}^D) \times \text{Comp}(X)]^{N(f)}$  with  $R((f_n, K_n))_{n=1}^{N(f)} = f$ .

Since  $\sigma \in C(\mathbb{R})$  satisfies the Kidger-Lyons conditions, then **Lemma 5.0.4** implies that there exists some  $\hat{c} \in \overset{\sigma}{\mathbf{NN}}_{d,N(f)}$  of width at most  $d + N(f) + 2$  for which, the associated the deep zero-sets  $\hat{K}_n \triangleq \{x \in X : I_{(\gamma,1]} \circ \sigma_{\text{sigmoid}} \circ \hat{c}(x)_n\}$  satisfy:

$$d_H(K_n, \hat{K}_n) < (2N(f))^{-1}\epsilon, \tag{12}$$

for each  $n = 1, \dots, N(f)$ . Since  $\sigma$  satisfies the Kidger-Lyons condition, then the universal approximation Theorem [Corollary 42] [37] implies that there are FFNNs  $\hat{f}_1, \dots, \hat{f}_{N(f)} \in \overset{\sigma}{\mathbf{NN}}_{d,D}$  of width at most  $d + D + 2$  satisfying the estimate:

$$\max_{n=1, \dots, N(f)} \|f_n - \hat{f}_n\|_{\infty} < (2N(f))^{-1}\epsilon. \tag{13}$$

Combining the estimates (12) and (13) with the estimate of **Proposition 3.2.1** yields:

$$\begin{aligned} D_{PC} \left( f \mid \sum_{n=1}^{N(f)} \hat{f}_n I_{\hat{K}_n} \right) &\leq \sum_{n=1}^{N(f)} \|f_n - \hat{f}_n\|_{\infty} + \sum_{n=1}^{N(f)} d_H(K_n, \hat{K}_n) \\ &\leq N(f)(2N(f))^{-1}\epsilon + N(f)(2N(f))^{-1}\epsilon \end{aligned} \tag{14} = \epsilon. \tag{15}$$

Thus, the result follows.  $\square$

#### C.4. Proof of Theorem 3.2.3

**Theorem 3.2.3** is a special case of the following, more detailed, result.

**Theorem 5.0.5** (Existence of an Optimal Partition: Extended Version). *Let  $L : \mathbb{R}^D \rightarrow \mathbb{R}$  be a continuous loss-function for which  $L(0) = 0$  and let  $\{\hat{f}_n\}_{n=1}^N \subseteq C(X, \mathbb{R}^D)$ . There exists a Borel measurable function  $C : \mathbb{R}^D \rightarrow \{n \leq N\}$  such that for each  $x \in \mathbb{R}^D$*

$$\min_{m \leq N} L(\hat{f}_m(x)) = L \left( \sum_{n \leq N} I_n(C(x)) \hat{f}_n^{\star}(x) \right), \tag{16}$$

where  $I_n : n = 1^N \rightarrow 0, 1$  and  $I_n(m) = 1$  if and only if  $n = m$ . Moreover, for each Borel probability measure  $\mathbb{P}$  on  $\mathbb{R}^d$  and each  $\delta \in (0, 1]$ , there exists a compact subset  $X_{\delta, \mathbb{P}} \subseteq \mathbb{R}^d$  such that:

- (i)  $\mathbb{P}(X_{\delta, \mathbb{P}}) \geq 1 - \delta$ ,
- (ii)  $C$  is continuous on  $X_{\delta, \mathbb{P}}$ ,
- (iii) For  $n \leq N$ ,  $K_n^{\star} \triangleq X_{\delta, \mathbb{P}} \cap C^{-1}\{n\}$  is a compact subset of  $X_{\delta, \mathbb{P}}$  on which

$$\min_{m \leq N} L(\hat{f}_m^{\star}(x)) = L(\hat{f}_n^{\star}(x)), \tag{17}$$

holds for each  $x \in K_n^{\star}$ .

- (iv)  $X_{\delta, \mathbb{P}} = \bigcup_{n \leq N} K_n^{\star}$  and for each  $n, m \leq N$  if  $n \neq m$  then  $K_m^{\star} \cap K_n^{\star} = \emptyset$ .
- (v) For each  $n \leq N$ ,  $K_n^{\star}$  is an open subset of  $X_{\delta, \mathbb{P}}$  (for its relative topology). In particular,  $X_{\delta, \mathbb{P}}$  is disconnected.

Moreover, if  $\mathbb{X} \subseteq X_{\delta, \mathbb{P}}$  and  $L(\hat{f}_n(x)) < \min_{n \neq \tilde{n}, \tilde{n} \leq N} L(\hat{f}_{\tilde{n}}(x))$  for each  $n \leq N$  and each  $x \in \mathbb{X}_n$  then for each  $n \leq N$

- $\mathbb{X}_n \subseteq K_n^{\star}$ ,
- $\mathbb{X}_n \cap K_m^{\star} = \emptyset$  for every  $m \leq N$  distinct from  $n$ .

**Proof** (Proof of **Theorem 5.0.5**). We begin by establishing the existence of a measurable selector, that is, a measurable function  $C$  satisfying (ii) on all of  $X$ . First, notice that the map  $(n_0, x) \mapsto L(\hat{f}_{n_0}(x))$ , from  $\{n \leq N\} \times \mathbb{R}^d$  to  $\mathbb{R}$  can be represented by

$$L(\hat{f}_{n_0}(x)) = \sum_{n_0 \leq N} I_{n_0}(n) L(\hat{f}_{n_0}(x)), \tag{18}$$

where  $I_{n_0}(n) = 1$  if  $n = n_0$  and 0 otherwise. Next, observe that the map  $x \mapsto \hat{f}_{n_0}(x)$  is continuous and this holds for each  $n_0 \leq N$ . Next, since  $L$  is continuous, each  $\hat{f}_{n_0}$  is continuous, and the composition of continuous functions is again continuous, then  $L \circ \hat{f}_{n_0}$  is continuous. Therefore, for each  $n_0 \leq N$  the map  $x \mapsto L(\hat{f}_{n_0}(x))$  is continuous from  $\mathbb{R}^d$  to  $\mathbb{R}$ . Next, observe that for any fixed  $x \in \mathbb{R}^d$ , the map  $n \mapsto \sum_{n_0 \leq N} I_{n_0}(n) L(\hat{f}_{n_0}(x))$  is a simple-function, thus, it is Borel-measurable. Hence, the map  $(n, x) \mapsto \sum_{n_0 \leq N} I_{n_0}(n) L(\hat{f}_{n_0}(x))$  is a Carathéodory function ([Definition 4.50] [83]).

Next, observe that the  $\Phi$  multi-function taking any  $x \in \mathbb{R}^d$  to the set  $\Phi(x) \triangleq \{n \leq N\}$  is constant. Therefore, it is a weakly-measurable correspondence in the sense of [Definition 18.1] [83]. Moreover, since the set  $\{n \leq N\}$  is a non-empty finite set then  $\Phi(x) = \{n \leq N\}$  is non-empty and compact for each  $x \in \mathbb{R}^d$ . Hence,  $\Phi$  is a weakly-measurable correspondence taking non-empty and compact-values in the separable metric space  $\mathbb{R}^d$ . Therefore, the conditions for the [Measurable Maximum Theorem; Theorem 18.19] [83] are met and thus, there exists some Borel-measurable function  $C : \mathbb{R}^d \rightarrow \{n \leq N\}$  satisfying (16)

$$\min_{n \leq N} \sum_{n_0 \leq N} I_{n_0}(n) L(\hat{f}_{n_0}(x)) = L(\hat{f}_{C(x)}(x)), \tag{19}$$

for every  $x \in \mathbb{R}^d$ . Since  $L(0) = 0$ , then, combining (18) and (19) we find that for every  $x \in \mathbb{R}^d$  the following holds (16) holds. This establishes the first claim.

Since  $\mathbb{P}$  is a Borel probability measure on  $X$  and since  $X$  is separable and metrizable then by [Theorem 13.6] [84]  $\mathbb{P}$  is a regular Borel measure on  $\mathbb{R}^d$ . Since  $\mathbb{R}^d$  is a metric space, it is a second-countable Hausdorff, and since  $\mathbb{R}^d$  is also locally-compact (since each  $x \in \mathbb{R}^d$  satisfies  $x \in \{z \in \mathbb{R}^d : \|x - z\| < 1\}$  and the latter is compact by the Heine-Borel theorem) then  $\mathbb{R}^d$  is a locally-compact Hausdorff space; thus, [Theorem 7.8] [85] applies therefore  $\mathbb{P}$  is a Radon measure on  $\mathbb{R}^d$ . Since  $\mathbb{P}$  is a regular Borel measure on  $\mathbb{R}^d$ ,  $\mathbb{R}^d$  is a locally-compact Hausdorff space,  $C$  is a Borel measurable function to  $\{n \leq N\}$ , and  $\{n \leq N\}$  is a separable metric space (when viewed as a metric sub-space of  $\mathbb{R}^N$ ), and  $\mathbb{P}(\mathbb{R}^d) = 1 < \infty$  since  $\mathbb{P}$  is a probability measure, then [Lusin's Theorem; Exercise 2.3.5] [86] applies; whence, for every  $\delta \in (0, 1]$  there exists some non-empty compact subset  $X_{\delta, \mathbb{P}} \subseteq \mathbb{R}^d$  for which  $\mathbb{P}(X_{\delta, \mathbb{P}}) \geq 1 - \delta$ . This establishes (i) and (ii).

Since  $C$  is continuous on  $X_{\delta, \mathbb{P}}$ , each  $\{n\}$  is closed in  $\{n_0 \leq N\}$ , since the pre-image of closed sets under continuous functions is again closed. Therefore, for each  $n \leq N$ ,  $C^{-1}\{n\}$  is a closed subset of  $\mathbb{R}^d$ . Since each  $K_n^{\star} = C^{-1}\{n\} \cap X_{\delta, \mathbb{P}}$  is a closed subset of  $X_{\delta, \mathbb{P}}$  and since  $X_{\delta, \mathbb{P}}$  is compact then each  $K_n^{\star}$  is compact. For each  $n \leq N$ ,  $x \in K_n^{\star}$  only if  $x \in C^{-1}\{n\}$ , only if  $C(x) = n$  and therefore condition:

$$L(\hat{f}(x)) = \min_{m \leq N} L(\hat{f}_m^{\star}(x)),$$

holds. This gives (17). Observe that

$$\begin{aligned} X_{\delta, \mathbb{P}} \cap \mathbb{R}^d &= X_{\delta, \mathbb{P}} \cap C^{-1}[\{n \leq N\}] = X_{\delta, \mathbb{P}} \cap \bigcup_{n \leq N} C^{-1}[\{n\}] \\ &= \bigcup_{n \leq N} X_{\delta, \mathbb{P}} \cap C^{-1}[\{n\}] = \bigcup_{n \leq N} K_n^{\star}. \end{aligned}$$

Therefore, (iii) holds.

Lastly, let  $n, m \leq N$  and  $n \neq m$ . Suppose that there exists some  $x \in K_n^{\star} \cap K_m^{\star}$ . Since  $x \in K_n^{\star}$  then  $C(x) = n$  and since  $x \in K_m^{\star}$  then  $C(x) = m$ ; therefore  $n = C(x) = m$  and  $n \neq m$ , a contradiction. Hence,  $K_n^{\star} \cap K_m^{\star} = \emptyset$  for each  $n, m \leq N$  with  $n \neq m$  and therefore (iv) holds.

Since  $\{n \leq N\}$  is a discrete metric space then for each  $n \leq N$ , the singleton set  $\{n\}$  is open in  $\{n \leq N\}$ . Therefore, by (ii),  $C$  is continuous on  $X_{\delta, \mathbb{P}}$  (for the relative topology) hence  $C^{-1}[\{n\}] \cap X_{\delta, \mathbb{P}}$  is open in  $X_{\delta, \mathbb{P}}$ . By (iv), pick  $1 < n \leq N$ , since  $K_n^{\star} \cap K_1^{\star} = \emptyset$  then [Definition 3.23] [25] is satisfied and therefore  $X_{\delta, \mathbb{P}}$  is not connected, i.e.: it is disconnected. This gives (v). If  $\mathbb{X} \subseteq X_{\delta, \mathbb{P}}$  then by (iv)  $\mathbb{X} \subseteq \bigcup_{n \leq N} K_n^{\star}$ . Since, for each  $n \leq N$  and each  $x \in \mathbb{X}_n$ ,  $L(\hat{f}_n(x)) < \min_{\bar{n} \neq n, \bar{n} \leq N} L(\hat{f}_{\bar{n}}(x))$  then  $\mathbb{X}_n \subseteq K_n^{\star}$  by (iii). Note, moreover, that by (iv)  $K_n^{\star} \cap K_m^{\star} = \emptyset$  for every  $n, m \leq N$  with  $n \neq m$  then there exists a unique  $n^x \in \{n \leq N\}$  such that  $x \in K_{n^x}^{\star}$  and therefore  $\mathbb{X}_n \cap K_m^{\star} = \emptyset$  for each  $m \leq N$  with  $m \neq n$ .  $\square$

### C.5. Proof of Proposition 3.2.3

**Proof** (Proof of Proposition 3.2.3). By construction, for each  $n \leq N$ ,  $K_n^{\star} \neq \emptyset$  and  $K_n^{\star} \cap \mathbb{X}_n \subseteq \emptyset$ . Therefore, there exists some  $\tilde{x} \in K_n^{\star} \cap \mathbb{X}_n$ . Since the set  $\{(x, y) \in \mathbb{X}^2 : x \neq y\}$  is a non-empty finite set then the minimum  $\min_{x, y \in \mathbb{X}, x \neq y} \|x - y\|$  is attained and it is non-zero since  $\|x - y\| > 0$  for each  $x, y \in \mathbb{X}$  with  $x \neq y$ . Thus,  $\Delta(\mathbb{X}) > 0$ . Therefore,  $\text{Ball}(\tilde{x}, \Delta(\mathbb{X})) \neq \emptyset$  and contained in  $K_n^{\star}$ . Therefore, by definition of the interior of  $K_n^{\star}$  we compute

$$\text{int}[K_n^{\star}] = \bigcup_{\substack{K_n^{\star} \subseteq U \subseteq \mathbb{R}^d \\ U \text{ is open}}} U \supseteq \text{Ball}(\tilde{x}, \Delta(\mathbb{X})) \neq \emptyset.$$

This gives (iii).

For (iv), note that by definition of  $\Delta(\mathbb{X})$ ,  $\text{Ball}(x, \Delta(\mathbb{X})) \cap \text{Ball}(y, \Delta(\mathbb{X})) = \emptyset$  for each  $x, y \in \mathbb{X}$  with  $x \neq y$  and by construction

$$K_n^{\star} = \bigcup_{x \in \mathbb{X}_n} \text{Ball}(x, \Delta(\mathbb{X})). \quad (20)$$

Therefore, for  $n, m \leq N$  with  $n \neq m$ , by construction  $\mathbb{X}_n \cap \mathbb{X}_m = \emptyset$  and therefore (20) implies that  $K_n^{\star} \cap K_m^{\star} = \emptyset$ .

For (i), note that, if  $f$  is of the form (1), then  $f = \sum_{n \leq N} f_n I_{\text{int}[K_n]}$ . Therefore, for  $x_1, x_2 \in \mathbb{X}$ , we have that there exists some  $n \leq N$  for which  $f(x_i) = f_n(x_i)$  for  $i = 1, 2$  if (but not only if)  $x_1, x_2 \in K_n^{\star}$ . Next, observe that by construction  $K_n^{\star} \cap \mathbb{X} = \mathbb{X}_n$ . Therefore,

$$\begin{aligned} \mathbb{P}((\exists n \leq N) f(x_i) = f_n(x_i) \text{ for } i = 1, 2) \\ \geq \mathbb{P}((\exists n \leq N) x_i \in K_n^{\star} \text{ for } i = 1, 2). \end{aligned} \quad (21)$$

The central results of [87] implies that

$$\begin{aligned} \mathbb{P}((\neg \exists n \leq N) (\exists n \leq N) x_i \in K_n^{\star} \text{ for } i = 1, 2) \\ \leq \frac{2^3}{\Delta(\mathbb{X})} \|x_1 - x_2\| \left( \sum_{i=\#\text{Ball}_{\mathbb{X}}(x_1, 2^{-3}\Delta(\mathbb{X}))}^{\#\text{Ball}_{\mathbb{X}}(x_1, \Delta(\mathbb{X}))} i^{-1} \right). \end{aligned} \quad (22)$$

For each positive integer  $I$ , since the partial sums of the harmonic series  $\sum_{i=1}^I i^{-1}$  are bounded above by  $\ln(I) + 1$ , then (22) implies that

$$\begin{aligned} \mathbb{P}((\neg \exists n \leq N) (\exists n \leq N) x_i \in K_n^{\star} \text{ for } i = 1, 2) \\ \leq \frac{2^3}{\Delta(\mathbb{X})} \|x_1 - x_2\| \left( \sum_{i=\#\text{Ball}_{\mathbb{X}}(x_1, 2^{-3}\Delta(\mathbb{X}))}^{\#\text{Ball}_{\mathbb{X}}(x_1, \Delta(\mathbb{X}))} i^{-1} \right) \\ \leq \frac{2^3}{\Delta(\mathbb{X})} \|x_1 - x_2\| \left( \sum_{i=1}^{\#\text{Ball}_{\mathbb{X}}(x_1, \Delta(\mathbb{X}))} i^{-1} \right) \\ \leq \frac{2^3}{\Delta(\mathbb{X})} \|x_1 - x_2\| \left( \sum_{i=1}^{\#\mathbb{X}} i^{-1} \right) \\ \leq \frac{2^3}{\Delta(\mathbb{X})} \|x_1 - x_2\| (\ln(\#\mathbb{X}) + 1). \end{aligned} \quad (23)$$

Combining (21) and (23) we obtain the following bound

$$\begin{aligned} \mathbb{P}((\exists n \leq N) f(x_i) = f_n(x_i) \text{ for } i = 1, 2) \\ \geq 1 - \frac{2^3}{\Delta(\mathbb{X})} \|x_1 - x_2\| (\ln(\#\mathbb{X}) + 1). \end{aligned} \quad (24)$$

Lastly, for (ii), by [65] and the definition of Subroutine 2 all but the last step of Subroutine 2 runs in polynomial time. The final step of Subroutine 2 runs in  $\mathcal{O}(N \log(N))$  since it is simply sorting procedure. Therefore Subroutine 2 terminates in polynomial time.  $\square$

## References

- [1] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, *Bull. Math. Biophys.* 5 (4) (1943) 115–133.
- [2] R. Cont, M.S. Müller, A stochastic partial differential equation model for limit order book dynamics, *SIAM J. Financial Math.* 12 (2) (2021) 744–787..
- [3] H. Buehler, L. Gonon, J. Teichmann, B. Wood, Deep hedging, *Quantitative Finance* 19 (8) (2019) 1271–1291.
- [4] C. Cuchiero, W. Khosrawi, J. Teichmann, A generative adversarial network approach to calibration of local stochastic volatility models, *Risks* 8 (4) (2020).
- [5] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014)..
- [6] P.J. Moore, T.J. Lyons, J. Gallacher, for the Alzheimer’s Disease Neuroimaging Initiative, Using path signatures to predict a diagnosis of alzheimer’s disease, *PLOS ONE* 14 (9) (2019) 1–16 (09 2019)..
- [7] A. Lapedes, R. Farber, Nonlinear signal processing using neural networks: Prediction and system modelling, IEEE international conference on neural networks (1987)..
- [8] R.G. Krishnan, U. Shalit, D. Sontag, Deep Kalman filters, *NeurIPS - Advances in Approximate Bayesian Inference* (2015)..
- [9] D. Rolnick, P.L. Donti, L.H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, A. Slavin Ross, N. Milojevic-Dupont, N. Jaques, A. Waldman-Brown, A. Luccioni, T. Maharaj, E.D. Sherwin, S. Karthik Muktavilli, K.P. Kording, C. Gomes, A.Y. Ng, D. Hassabis, J.C. Platt, F. Creutzig, J. Chayes, Y. Bengio, Tackling climate change with machine learning, arXiv e-prints (Jun. 2019)..
- [10] K. Hornik, M. Stinchcombe, H. White, Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks, *Neural Netw.* 3 (5) (1990) 551–560 (Oct. 1990)..
- [11] A.R. Barron, Universal approximation bounds for superpositions of a sigmoidal function, *IEEE Trans. Inform. Theory* 39 (3) (1993) 930–945.
- [12] T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, Q. Liao, Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review, *Int. J. Autom. Comput.* 14 (5) (2017) 503–519 (2017)..
- [13] D. Yarotsky, A. Zheverchuk, The phase diagram of approximation rates for deep neural networks, in: *Advances in Neural Information Processing Systems*, Vol. 33, Curran Associates Inc, 2020, pp. 13005–13015 (2020)..
- [14] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (2015)..
- [15] Q. Li, L. Chen, C. Tai, W. E. Maximum principle based algorithms for deep learning, *Journal of Machine Learning Research* 18 (165) (2018) 1–29 (2018)..
- [16] A.G. Baydin, B.A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: A survey 18 (1) (2017) 5595–5637 (Jan. 2017)..
- [17] A. Patrascu, P. Irofti, Stochastic proximal splitting algorithm for composite minimization, *Optimiz. Lett.* 2021 (2021) 1–19.
- [18] A. Bietti, J. Mairal, On the inductive bias of neural tangent kernels, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Vol. 32, Curran Associates Inc, 2019 (2019)..

- [19] J. Heiss, J. Teichmann, H. Wutte, How implicit regularization of neural networks affects the learned function—part i, arXiv preprint arXiv:1911.02903 (2019)..
- [20] G. Cybenko, Approximation by superpositions of a sigmoid function, *Mathematics of Control, Signals Syst.* 2 (4) (1989) 303–314.
- [21] K.-I. Funahashi, On the approximate realization of continuous mappings by neural networks, *Neural Networks* 2 (3) (1989) 183–192.
- [22] K. Hornik, Approximation capabilities of multilayer feedforward networks, *Neural Networks* 4 (2) (1991) 251–257 (1991). doi: 10.1016/0893-6080(91)90009-T..
- [23] S.S. Lee, P.A. Mykland, Jumps in Financial Markets: A New Nonparametric Test and Jump Dynamics, *The Review of Financial Studies* 21 (6) (2007) 2535–2563 (12 2007)..
- [24] R. Cont, P. Tankov, Financial modelling with jump processes, *Chapman & Hall/CRC Financial Mathematics Series*, Chapman & Hall/CRC, Boca Raton, FL, 2004 (2004).
- [25] J.R. Munkres, *Topology*, Prentice Hall Inc, Upper Saddle River, NJ, 2000, second edition of [MR0464128] (2000)..
- [26] R. Gribonval, G. Kutyniok, M. Nielsen, F. Voigtlaender, Approximation spaces of deep neural networks, *Constructive Approximation* 2021 (2021) 1–109.
- [27] J.W. Siegel, J. Xu, Approximation rates for neural networks with general activation functions, *Neural Networks* 128 (2020) (2020) 313–321.
- [28] A. Kratsios, The universal approximation property, *Ann. Math. Artif. Intell.* (01 2021)..
- [29] D.A. Findlay, Training networks with discontinuous activation functions, in: 1989 First IEE International Conference on Artificial Neural Networks, 1989, pp. 361–363.
- [30] L.V. Ferreira, E. Kaszkurewicz, A. Bhaya, Solving systems of linear equations via gradient systems with discontinuous righthand sides: application to ls-svm, *IEEE Trans. Neural Networks* 16 (2) (2005, 2005,) 501–505.
- [31] G.-B. Huang, Q.-Y. Zhu, K. Mao, C.-K. Siew, P. Saratchandran, N. Sundararajan, Can threshold networks be trained directly?, *IEEE Trans. Circuits Syst. II: Express Briefs* 53 (3) (2006, 2006,) 187–191.
- [32] P. Kidger, T. Lyons, Universal approximation with deep narrow networks, *International Conference on Learning Theory 2020* (2020) 2306–2327.
- [33] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, Graph neural networks: A review of methods and applications, *AI Open* 1 (2020) (2020) 57–81.
- [34] M.M. Bronstein, J. Bruna, T. Cohen, P. Velicković, Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges, arXiv e-prints (2021) arXiv:2104.13478 (Apr. 2021)..
- [35] O. Ganea, G. Bécigneul, T. Hofmann, Hyperbolic neural networks, *Adv. Neural Inform. Processing Syst.* 2018 (2018) 5345–5355.
- [36] A. Kratsios, I. Bilokopytov, Non-euclidean universal approximation, in: *Advances in Neural Information Processing Systems*, Vol. 33, Curran Associates Inc, 2020, pp. 10635–10646 (2020)..
- [37] A. Kratsios, L. Papon, Universal approximation theorems for differentiable geometric deep learning, arXiv e-prints (jan 2021)..
- [38] A. Zamanlooy, Behnoosh Kratsios, T. Liu, I. Dokmanić, Universal Approximation Under Constraints is Possible with Transformers, arXiv e-prints (2021) arXiv:2110.03303 (Oct. 2021). arXiv:2110.03303..
- [39] T. Cohen, M. Weiler, B. Kicanaoglu, M. Welling, Gauge equivariant convolutional networks and the icosahedral CNN, in: K. Chaudhuri, R. Salakhutdinov (Eds.), *Proceedings of the 36th International Conference on Machine Learning*, Vol. 97 of Proceedings of Machine Learning Research, PMLR, 2019, pp. 1321–1330 (09–15 Jun 2019)..
- [40] P. Petersen, F. Voigtlaender, Equivalence of approximation by convolutional neural networks and fully-connected networks, *Proc. Am. Math. Soc.* 148 (4) (2020) 1567–1581.
- [41] D. Yarotsky, Universal approximations of invariant maps by neural networks, *Constructive Approximation* 2021 (2021) 1–68.
- [42] C. Durkan, A. Bekasov, I. Murray, G. Papamakarios, Neural spline flows, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Vol. 32, Curran Associates Inc, 2019 (2019)..
- [43] R.T.Q. Chen, Y. Rubanova, J. Bettencourt, D.K. Duvenaud, Neural ordinary differential equations, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Vol. 31, Curran Associates Inc, 2018 (2018). URL:https://proceedings.neurips.cc/paper/2018/file/69386f6bb1dfed68692a24c8686939b9-Paper.pdf..
- [44] W. Grathwohl, R.T.Q. Chen, J. Bettencourt, D. Duvenaud, Scalable reversible generative models with free-form continuous dynamics, in: *International Conference on Learning Representations*, 2019 (2019)..
- [45] A. Kratsios, C. Hyndman, Neu: A meta-algorithm for universal uap-invariant feature representation, *J. Mach. Learn. Res.* 22 (92) (2021) 1–51.
- [46] A. Faragó, G. Lugosi, Strong universal consistency of neural network classifiers, *IEEE Trans. Inform. Theory* 39 (4) (1993) 1146–1151.
- [47] J.-P. Aubin, H. Frankowska, *Set-valued analysis*, Modern Birkhäuser Classics, Birkhäuser Boston Inc, Boston, MA, 2009, reprint of the 1990 edition [MR1048347] (2009)..
- [48] D. Yarotsky, Optimal approximation of continuous functions by very deep relu networks, in: *Proceedings of the 31st Conference On Learning Theory*, Vol. 75 of Proceedings of Machine Learning Research, PMLR, 2018, pp. 639–649 (06–09 Jul 2018)..
- [49] I. Petrakakis, A direct constructive proof of a Stone-Weierstrass theorem for metric spaces, in: *Pursuit of the universal*, Vol. 9709 of Lecture Notes in Comput. Sci., Springer, [Cham], 2016, pp. 364–374 (2016)..
- [50] Z. Lu, H. Pu, F. Wang, Z. Hu, L. Wang, The expressive power of neural networks: A view from the width, *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 6232–6240.
- [51] J.W. Siegel, J. Xu, Approximation rates for neural networks with general activation functions, *Neural Networks* (2020).
- [52] Á. Capel, J. Ocariz, Approximation with neural networks in variable lebesgue spaces, arXiv preprint arXiv:2007.04166 (2020)..
- [53] P. Cheridito, A. Jentzen, F. Rossmannek, Efficient approximation of high-dimensional functions with neural networks, *IEEE Trans. Neural Networks Learn. Syst.* (2021).
- [54] I. Gühring, M. Raslan, Approximation rates for neural networks with encodable weights in smoothness spaces, *Neural Networks* 134 (2021) (2021) 107–130.
- [55] D. Burago, Y. Burago, S. Ivanov, *A course in metric geometry*, Graduate Studies in Mathematics, vol. 33, American Mathematical Society, Providence, RI, 2001.
- [56] S. Si, D. Tao, B. Geng, Bregman divergence-based regularization for transfer subspace learning, *IEEE Trans. Knowl. Data Eng.* 22 (7) (2010) 929–942.
- [57] A. Banerjee, S. Merugu, I.S. Dhillon, J. Ghosh, Clustering with Bregman divergences, *J. Mach. Learn. Res.* 6 (2005) (2005) 1705–1749.
- [58] B.A. Frigiyik, S. Srivastava, M.R. Gupta, Functional Bregman divergence and Bayesian estimation of distributions, *IEEE Trans. Inform. Theory* 54 (11) (2008) 5130–5139.
- [59] S. Kullback, R.A. Leibler, On information and sufficiency, *Ann. Math. Statistics* 22 (1951) (1951) 79–86.
- [60] S.-I. Amari, H. Nagaoka, *Methods of information geometry*, *Am. Math. Soc.* 191 (2000).
- [61] R. Agrawal, T. Horel, Optimal bounds between f-divergences and integral probability metrics, *J. Mach. Learn. Res.* 22 (128) (2021) 1–59.
- [62] F.H. Clarke, Generalized gradients and applications, *Trans. Amer. Math. Soc.* 205 (1975) 247–262.
- [63] E. Dahlhaus, D.S. Johnson, C.H. Papadimitriou, P.D. Seymour, M. Yannakakis, The complexity of multiterminal cuts, *SIAM J. Comput.* 23 (4) (1994) 864–894.
- [64] P.S. Pulat, On the relation of max-flow to min-cut for generalized networks, *European. J Oper. Res.* 39 (1) (1989) 103–107.
- [65] Y. Bartal, On approximating arbitrary metrics by tree metrics, in: *STOC '98* (Dallas, TX), ACM, New York, 1999, pp. 161–168 (1999)..
- [66] S.J. Reddi, S. Kale, S. Kumar, On the convergence of adam and beyond (2018)..
- [67] E. Gelenbe, Stability of the random neural network model, *Neural Computation* 2 (2) (1990) 239–247.
- [68] C. Louart, Z. Liao, R. Couillet, et al., A random matrix approach to neural networks, *Ann. Appl. Prob.* 28 (2) (2018) 1190–1248.
- [69] G. Yehudai, O. Shamir, On the power and limitations of random features for understanding neural networks, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Vol. 32, Curran Associates Inc, 2019 (2019)..
- [70] C. Cuchiero, M. Larsson, J. Teichmann, Deep neural networks, generic universal interpolation, and controlled odes, *SIAM J. Math. Data Sci.* 2 (3) (2020) 901–919.
- [71] L. Gonon, L. Grigoryeva, J. Ortega, Approximation bounds for random neural networks and reservoir systems, *ArXiv abs/2002.05933* (2020).
- [72] R. Cont, P. Tankov, Financial modelling with jump processes, *Chapman & Hall/CRC Financial Mathematics Series*, Chapman & Hall/CRC, Boca Raton, FL, 2004 (2004)..
- [73] D. Filipović, M. Larsson, Polynomial jump-diffusion models, *Stoch. Syst.* 10 (1) (2020) 71–97 (2020)..
- [74] J. Johnson, Deep, skinny neural networks are not universal approximators, arXiv preprint arXiv:1810.00393 (2018)..
- [75] S. Park, C. Yun, J. Lee, J. Shin, Minimum width for universal approximation, arXiv preprint arXiv:2006.08859 (2020)..
- [76] Kaggle, California housing prices, URL:https://www.kaggle.com/camnugent/california-housing-prices, accessed: 2020-05-15 (2017)..
- [77] A. Geron, handson-ml, URL:https://github.com/ageron/handson-ml/tree/master/datasets/housing, accessed: 2020-05-15 (2018)..
- [78] L. de Haan, A. Ferreira, *Extreme value theory*, Springer Series in Operations Research and Financial Engineering, Springer, New York, 2006, an introduction.
- [79] L. Narici, E. Beckenstein, *Topological vector spaces*, 2nd Edition, Vol. 296 of Pure and Applied Mathematics (Boca Raton), CRC Press, Boca Raton, FL, 2011 (2011)..
- [80] A. Caragea, P. Petersen, F. Voigtlaender, Neural network approximation and estimation of classifiers with classification boundary in a barron class, arXiv preprint arXiv:2011.09363 (2020)..
- [81] D. Pallaschke, D. Pumplün, Spaces of Lipschitz functions on metric spaces, *Discuss. Math. Differ. Incl. Control Optim.* 35 (1) (2015) 5–23 (2015)..
- [82] H. Hoffmann, On the continuity of the inverses of strictly monotonic functions, *Irish Math. Soc. Bull.* 1 (75) (2015) 45–57 (2015)..
- [83] C.D. Aliprantis, K.C. Border, *Infinite dimensional analysis, third ed.*, Springer, Berlin, 2006, a hitchhiker's guide.
- [84] A. Klenke, *Probability theory, second ed.*, Universitext, Springer, London, 2014, a comprehensive course (2014).

- [85] G.B. Folland, *Real analysis: modern techniques and their applications*, John Wiley & Sons 2013 (2013).
- [86] H. Federer, *Geometric measure theory*, Die Grundlehren der mathematischen Wissenschaften, Band 153, Springer-Verlag, New York Inc., New York, 1969 (1969)..
- [87] G. Calinescu, H. Karloff, Y. Rabani, Approximation algorithms for the 0-extension problem, *SIAM J. Comput.* 34 (2) (2004/05) 358–372 (2004/05)..



**Anastasis Kratsios** Anastasis received his PhD in 2018 in mathematics. He pursued a postdoctoral fellowship at ETH Zürich in 2018–2021 in the mathematics department researching non-Euclidean universal approximation and the foundations of geometric deep learning. He is currently a researcher at the university of Basel working on geometric deep learning, developing new architectures for structured learning, and how such tools can be deployed to approximate functions taking values in manifold, graphs, spaces of probability measures, and non-linear spaces of stochastic processes.



**Behnoosh Zamanlooy** Since September 2020, Behnoosh is a Ph.D. student advised by Prof. Sven Seuken in the Computation and Economics Research Group at the Department of Informatics of the University of Zürich. Behnoosh holds a Master's degree in Mathematical Finance from Concordia University, Canada, and a Bachelor's degree in Computer Science from University of Tehran, Iran. She worked as a Quantitative Researcher from February 2019 for a year and then was promoted to the Director of quantitative researcher, modeling of electricity prices and derivatives, at a Montréal private investment firm.