Semi-supervised Conditional Density Estimation with Wasserstein Laplacian Regularisation

Olivier Graffeuille[1], **Yun Sing Koh**[1], **Jörg Wicker**[1], **Moritz Lehmann**[2]
[1] School of Computer Science, The University of Auckland
[2] Xerra Earth Observation Institute, The University of Waikato
ogra439@aucklanduni.ac.nz, y.koh@auckland.ac.nz, j.wicker@auckland.ac.nz, mlehmann@waikato.ac.nz

Abstract

Conditional Density Estimation (CDE) has wide-reaching applicability to various real-world problems, such as spatial density estimation and environmental modelling. CDE estimates the probability density of a random variable rather than a single value and can thus model uncertainty and inverse problems. This task is inherently more complex than regression, and many algorithms suffer from overfitting, particularly when modelled with few labelled data points. For applications where unlabelled data is abundant but labelled data is scarce, we propose Wasserstein Laplacian Regularisation, a semi-supervised learning framework that allows CDE algorithms to leverage these unlabelled data. The framework minimises an objective function which ensures that the learned model is smooth along the manifold of the underlying data, as measured by Wasserstein distance. When applying our framework to Mixture Density Networks, the resulting semi-supervised algorithm can achieve similar performance to a supervised model with up to three times as many labelled data points on baseline datasets. We additionally apply our technique to the problem of remote sensing for chlorophyll-a estimation in inland waters.

Introduction

Conditional Density Estimation (CDE) is the task of estimating the probability density $p(y|x)$ of a random continuous variable y given an input x, as opposed to regression where we estimate a single value $f(y|x)$. Approaches to CDE include Mixture Density Networks (MDNs) (Bishop 1994), discretised histograms (Van Oord, Kalchbrenner, and Kavukcuoglu 2016), Normalising Flows (NFs) (Trippe and Turner 2018) and the Uncountable Mixture of Asymmetric Laplacians (UMAL) model (Brando et al. 2019). By producing a probability density around a response variable, CDE allows practitioners to make informed decisions relating to the range of predicted outcomes. Another benefit of CDE is the ability to model *inverse problems*. Inverse problems estimate the causal agents that lead to an observed state in a response variable (Bishop 2006) and are often ill-posed as they do not have a unique solution for a given observation. By generating a probability density, CDE is able to model this non-uniqueness, as shown in Figure 1. However, CDE is inherently a more complex task than regression and many CDE techniques suffer from overfitting and require large datasets.

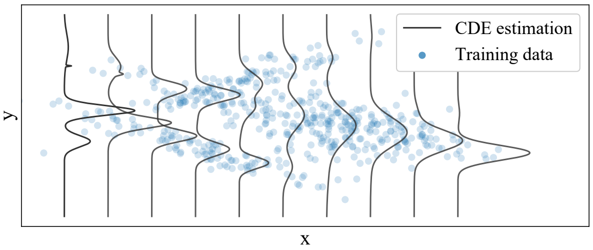


Figure 1: CDE model learns ill-posed relationship between x and y in 1D synthetic Gaussians dataset, by estimating distributions of y conditioned on x.

Many applications lack labelled data, thus it is desirable to use methods that exploit unlabelled data to improve learning (Levatić et al. 2020). Semi-Supervised Learning (SSL) approaches make assumptions about the relationship between the underlying data distribution and the output variable to gain information from unlabelled data (Luo et al. 2017; Li, Zha, and Zhou 2017). We hypothesise that by leveraging unlabelled data, we can use regularisation to reduce overfitting and improve the performance of CDE algorithms.

Currently, some deep generative models for conditional distributions (Mirza and Osindero 2014; Sohn, Lee, and Yan 2015) use unlabelled data to improve performance (Shu, Bui, and Ghavamzadeh 2017). However, these models require structured outputs and are unable to model univariate or low-dimensional outputs, which is restrictive to real-world applicability. Additionally, these models cannot estimate tractable densities, which makes them less informative. Other related work includes Huang (2021), who used normalising flows to estimate conditional densities of missing features and perform imputation and classification simultaneously in a SSL setting. Khan and Sugiyama (2012) extended Least Squares CDE to learn from unlabelled data, however this approach does not have the expressive power of deep learning CDE techniques (Sugiyama et al. 2010).

To bridge this gap, our research aims to leverage unlabelled data to improve performance of CDE techniques. Intuition of our approach can be gained by looking to the *smoothness assumption* of SSL, which uses unlabelled data to smooth models under the assumption that if two points x_1, x_2 in a high-density region are close then their corresponding outputs $f(y|x_1), f(y|x_2)$ should also be close. This idea is extensively used in regression settings, but intuitively can equivalently be applied to CDE by ensuring that the output probability densities $p(y|x_1), p(y|x_2)$ are similar.

Our proposed Wasserstein Laplacian Regularisation (WLR) framework adds an unsupervised objective function during training of supervised CDE algorithms to ensure that they learn a *smooth* function. The WLR objective function works by penalising difference in output probability densities of similar training points. Difference in output densities is measured by *Wasserstein distance*. We demonstrate the effectiveness of our framework by implementing it in MDNs.

Many CDE applications could potentially benefit from leveraging unlabelled data including rainfall-runoff modelling (Klotz et al. 2021), forest fire risk assessment (Bisquert et al. 2012), spatial density modelling (Trippe and Turner 2018) and remote sensing chlorophyll-a (Pahlevan et al. 2020). As a case study we use remote sensing of chlorophyll-a. Specifically, using multispectral data from satellites to estimate the concentration of chlorophyll-a pigment in inland waters, for monitoring of harmful algal blooms. Ground truth measurements require a manual sample collection and lab analysis, which is expensive. While these labelled data are limited, the unlabelled satellite data is abundant. Learning the relationship between multispectral data and chlorophyll-a concentration is difficult. A specific multispectral signal could be produced by different concentrations of chlorophyll-a, depending on the other water constituents. This is thus an ill-posed problem, and the current state-of-the-art approach in this field uses CDE to overcome this (Pahlevan et al. 2020). CDE is beneficial because a probability density estimation of chlorophyll-a pigments and is more informative than a single value to practitioners for risk assessment. Current approaches to remote sensing chlorophyll-a do not make use of the unlabelled data.

Our contributions are three-fold. (1) We introduce the first semi-supervised technique for conditional density estimation with deep learning, by proposing a framework which allows conditional density estimation algorithms to leverage unlabelled data during training. (2) We apply our framework to MDNs, and show that the resulting semi-supervised algorithm with fewer labelled data points performs equally to a supervised algorithm with more labelled data points in terms of Negative Log Likelihood. (3) We demonstrate the utility of our technique in a real-world context with a case study on remote-sensing chlorophyll-a.

The remainder of the paper is structured as follows. Section establishes preliminary concepts. In Section , we propose our semi-supervised CDE framework. Section covers experimental results. Section concludes the paper.

Preliminaries

SSL and Laplacian Regularisation

The task of SSL can be described as the learning of a function $f : \mathcal{X} \mapsto \mathcal{Y}$ using a training set of l labelled examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$ and u unlabelled examples $\{(\boldsymbol{x}_i)\}_{i=l+1}^{l+u}$ with data points $\boldsymbol{x} \in \mathcal{X}$ and labels $y \in \mathcal{Y}$.

Laplacian Regularisation is a semi-supervised regularisation framework which smooths a function by minimising the gradient of the function along a manifold (Belkin, Niyogi, and Sindhwani 2006). This is often formulated as an objective function which minimises the difference in function outputs given similar inputs:

$$\min_f \sum_{i=1}^{l+u} \sum_{j=1}^{l+u} w_{ij} L_p(f(\boldsymbol{x}_i), f(\boldsymbol{x}_j)) \tag{1}$$

where $f(.)$ is the function being smoothed, L_p is a loss function such as L_2, and w_{ij} is a pairwise weight defined by a graph Laplacian. Here w causes nearby points to be more greatly penalised for a difference in output, and is generally calculated by a kNN matrix and a kernel function, the most common of which is the Gaussian Radial Basis Function (RBF):

$$w_{ij} = a_{ij} K_{RBF}^{d}(\boldsymbol{x}_i, \boldsymbol{x}_j) = a_{ij} \exp\left(-\frac{|\boldsymbol{x}_i - \boldsymbol{x}_j|^d}{2\sigma^d}\right) \tag{2}$$

where $a_{ij} = 1$ if points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are neighbours under kNN, and 0 otherwise.

CDE and Mixture Density Networks

Performance of CDE is generally measured by the mean Negative Log Likelihood (NLL) of the model's estimated Probability Density Function (PDF) $f(.)$ given the real data y. Thus, NLL is directly correlated to the likelihood of test data on our model. Henceforth, when talking about performance of CDE models, we refer to NLL.

$$NLL(f(.), y) = -\ln(f(y)) \tag{3}$$

MDNs are a type of feedforward neural network which outputs a vector of parameters that define a mixture distribution as an estimate, rather than a single value prediction, in order to perform CDE (Bishop 1994). In this paper, the output of MDN models will parameterise a Gaussian mixture PDF. One such mixture PDF $f_i(.)$ has c components and parameters $\{\mu_{ij}, \sigma_{ij}, \pi_{ij}\}_{j=1}^{c}$, which are defined by MDN model $g(.)$ with parameters θ given input $\boldsymbol{x}_i$:

$$\begin{gathered} f_i(.) = p(y|\boldsymbol{x}_i) = \sum_{j=1}^{c} \pi_{ij} \mathcal{N}(y|\mu_{ij}, \sigma_{ij}) \\ \{\mu_{ij}, \sigma_{ij}, \pi_{ij}\}_{j=1}^{c} = g(\boldsymbol{x}_i|\theta) \end{gathered} \tag{4}$$

MDNs are trained by gradient descent with NLL as objective function:

$$\min_\theta \frac{1}{l} \sum_{i=1}^{l} NLL(f_i(.), y_i) \tag{5}$$

Wasserstein Distance

Given two distributions f_1 and f_2, 1-Wasserstein distance is defined as:

$$\mathcal{W}_1(f_1, f_2) = \inf_{\gamma \in \Pi(f_1, f_2)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||] \tag{6}$$

where $\Pi(f_1, f_2)$ is the set of all joint distributions of $\gamma(x, y)$ whose marginals are respectively f_1, f_2. Intuitively, the 1-Wasserstein distance, known as the Earth Mover's Distance, can be described as the amount of "earth" that must be transported to transform f_1 into f_2.

Methodology

Laplacian Regularisation improves model performance by penalising the difference in model outputs given similar inputs. This framework relies on two fundamental assumptions of semi-supervised learning: the *smoothness* assumption that two similar inputs should produce a similar output, and the *manifold* assumption that the underlying data lie approximately on a manifold of lower dimension than the input space. Our work begins with the observation that these assumptions are also sensible for the task of CDE, where we are estimating a probability density. In the context of chlorophyll-a estimation, this smoothness would mean that two multispectral inputs of similar colour would estimate similar chlorophyll-a concentration probability densities.

To apply these ideas to CDE, we first need to define the similarity of outputs, specifically a similarity of probability densities. We use Wasserstein distance as our measure of similarity, due to properties discussed in this section which make it suitable for use as an objective function. Our proposed framework for semi-supervised CDE, Wasserstein Laplacian Regularisation, implements an unsupervised objective function during the training of CDE algorithms. This objective function allows the model to learn a smooth function by penalising the Wasserstein distance between estimated probability densities of similar inputs. WLR improves performance for two reasons. Firstly, the regularisation prevents overfitting, which is common in CDE approaches, particularly with few labelled data. Secondly, it ensures that the function learnt by the model is a reasonable prediction in regions without labelled data by effectively interpolating predictions of labelled data points to nearby unlabelled data. This framework can be applied to any CDE algorithm which is trained via an optimisation framework. To show its utility, we apply WLR to Mixture Density Networks. We selected MDNs as the basis of our framework instead of non-parametric CDE algorithms such as NFs, UMAL or discretised histograms, as they achieved the best performance in supervised experiments with few labelled data points. Our resulting semi-supervised algorithm, Wasserstein MDN (WMDN), minimise the WLR objective function as well as NLL during training by gradient descent, as shown in Figure 2.

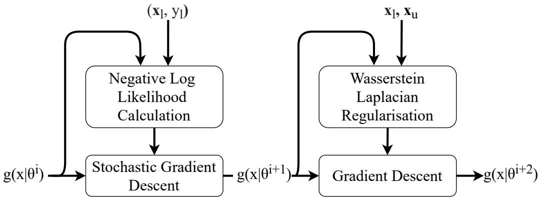

$(\mathbf{x}_l, y_l)$
Negative Log Likelihood Calculation
$g(x|\theta^i)$
Stochastic Gradient Descent
$g(x|\theta^{i+1})$
$\mathbf{x}_l, \mathbf{x}_u$
Wasserstein Laplacian Regularisation
Gradient Descent
$g(x|\theta^{i+2})$

Figure 2: Training process of WMDN $g(.)$ with weights θ over one epoch, with NLL and WLR objective functions.

Wasserstein Distance as an Objective Function

Unlike most statistical distances, Wasserstein distance captures not just the difference in magnitude between distributions but the metric space or "ground distance" between them. Based on the objective function, this property results in informative gradients even when the distributions are not overlapping (Arjovsky, Chintala, and Bottou 2017). Wasserstein is also symmetric ($\mathcal{W}(f_1, f_2) = \mathcal{W}(f_2, f_1)$), is always finite, and is continuous given a continuous change in inputs (Arjovsky, Chintala, and Bottou 2017). Finally, when used to interpolate between distributions, Wasserstein distance has the advantage of generating sensible barycentres which preserve distribution spread (Solomon et al. 2014). Given these properties, Wasserstein distance is a suitable objective function for our regularisation framework:

$$\min_f \sum_{i=1}^{l+u} \sum_{j=1}^{l+u} w_{ij} \mathcal{W}_p^q(f_i(.), f_j(.)) \tag{7}$$

where $\mathcal{W}_p(.)$ is the univariate p-Wasserstein distance, $f_i(.), f_j(.)$ are PDFs parameterised by $f(x_i), f(x_j)$ respectively as in Equation 4, w is a weight defined in Equation 2, and q is a power parameter analogous to L_p loss. Intuitively, this term will be smaller if the function f is smoother; the term will be zero if the model estimates the same probability density for any input.

Wasserstein Distance Computation

Although Wasserstein distance has advantageous properties for an objective function, it is both expensive to compute and not suitable for gradient backpropagation. In our framework, we approximate the 1-Wasserstein distance by discretising $f_i(.), f_j(.)$ at b linearly spaced values with interval s (Sakai 2018):

$$\begin{gathered} \boldsymbol{y} = \{y_n | y_n = y_0 + ns, n = 0, 1, \ldots, b-1\} \\ \mathcal{W}_1(f_i(.), f_j(.)) \approx s \sum_{n=0}^{b} |F_i(y_n) - F_j(y_n)| \end{gathered} \tag{8}$$

where $F_i(.)$ is the discretised Cumulative Density Function (CDF) of the discretised PDF $f_i(.)$:

$$F_i(y_n) = s \sum_{k=0}^{n} f_i(y_k) \tag{9}$$

where the s term normalises the CDF such that $\sum F_i \approx 1$. Normalisation could also be achieved by dividing F_i by its sum, but we observe experimentally that this results in gradients which are unstable with small changes to f_i. As $b \to \infty$, the error in Equation 8 approaches zero. This form is not exact and restricts us to $\mathcal{W}_1$. However, it is computationally simpler and usable for backpropagation. Combining Equations 7, 8 and 9, the WLR objective function is:

$$\min_f \sum_{i=1}^{l+u} \sum_{j=1}^{l+u} w_{ij} \left(s^2 \sum_{n=0}^{b} \left| \sum_{k=0}^{n} f_i(y_k) - f_j(y_n) \right| \right)^q \tag{10}$$

The theoretical runtime of this objective function is $\mathcal{O}(nkb^2)$, where $n = l + u$ is the number of training data points, and k parameterises kNN in Equation 2.

Algorithm 1: WMDN training scheme

```
Input: $X_l, Y_l, X_v, Y_v, X_u$
Parameter: $\eta, q, d, b, \lambda_u, k$
Output: trained MDN $g(.)$
 1: Initialise MDN $g(.)$ with parameters $\theta$
 2: $l, u, v \leftarrow |X_l|, |X_u|, |X_v|$
 3: $\sigma \leftarrow$ median($\{\min_{\boldsymbol{x} \in X_l \cup X_u \setminus \boldsymbol{x}_i} \|\boldsymbol{x}_i - \boldsymbol{x}\|^d | \boldsymbol{x}_i \in X_l \cup X_u\}$)    ▷ Median 1NN distance
 4: while true do
 5:     for $(X_{batch}, Y_{batch}) \subseteq (X_l, Y_l)$ do
 6:         $L_{NLL} \leftarrow \frac{1}{|X_{batch}|} \sum_{x_i \in X_{batch}} NLL(f_i(.), y_i)$
 7:         $\theta \leftarrow \theta - \eta \frac{\delta\theta}{\delta L_{NLL}}$    ▷ SGD
 8:     end for
 9:     $w_{ij} \leftarrow a_{ij} exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^d}{2\sigma^d}\right)$    ▷ $a_{ij}$ from kNN
10:     $L_W \leftarrow \frac{\lambda_u}{(l+u)^2} \sum_{i=0}^{l+u} \sum_{j=0}^{l+u} w_{ij} \mathcal{W}_1^q(f_i(.), f_j(.), b)$
11:     $\theta \leftarrow \theta - \eta \frac{\delta\theta}{\delta L_W}$    ▷ GD
12:     $L_v \leftarrow \frac{1}{v} \sum_{i=0}^{v} NLL(f_i(.), y_i)$
13:     if $L_v < L_v^{previous}$ then
14:         break    ▷ Stop if validation error increases
15:     end if
16: end while
```

Algorithm 2: $\mathcal{W}_1$ function

```
Input: $f_1, f_2, b$    ▷ PDFs to compare, number of bins
Output: $\mathcal{W}_1$    ▷ 1-Wasserstein distance
$s \leftarrow \frac{\max(y_l) - \min(y_l)}{b-1}$    ▷ Bin stepsize
$\boldsymbol{y} = \{y_n | y_n = \min(y_l) + ns, n = 0, 1, \ldots, b-1\}$
$\boldsymbol{F_1} \leftarrow [\sum_{j=0}^{i} f_1(y_j)$ for $i = 1, 2, \ldots, b]$    ▷ CDF
$\boldsymbol{F_2} \leftarrow [\sum_{j=0}^{i} f_2(y_j)$ for $i = 1, 2, \ldots, b]$
$\mathcal{W}_1 \leftarrow s^2 \sum_{i=0}^{b} |\boldsymbol{F_1} - \boldsymbol{F_2}|$
```

Wasserstein Mixture Density Networks

Here we describe our WMDN, which is identical to MDN, except that it minimises the WLR objective function as well as NLL during training:

$$\min_{\theta} \frac{1}{l} \sum_{i=1}^{l} NLL(f_i(.), y_i) + \frac{\gamma_u}{(l+u)^2} \sum_{i=1}^{l+u} \sum_{j=1}^{l+u} w_{ij} \mathcal{W}_1^q(f_i(.), f_j(.)) \tag{11}$$

where θ are the parameters of the WMDN model $g(.)$ we are training, γ_u is the SSL weight coefficient and q is a power parameter analogous to L_p loss. Figure 2 shows each epoch of the WMDN training, the supervised loss term is applied in batches, while the unsupervised loss term is applied to all training data in a single batch. Maximising the number of points used in the WLR objective will improve the smoothness of the regularisation. The pseudocode for the WMDN is provided in Algorithms 1 and 2; Lines 9 to 11 in Algorithm 1 describe the WLR framework.

Experimental Results

We evaluate the effectiveness of the WLR framework by evaluating and comparing the performance of MDN and our proposed WMDN. All experiments are implemented in PyTorch (Paszke et al. 2019), running on a GeForce RTX 3080 GPU. A full list of algorithm parameters, datasets, and source code are available online[1]. Regarding WMDN parameters, learning rate η and SSL weight coefficient λ_u were tuned for each dataset, while other parameters were fixed and selected as values that performed well across datasets in typical validation/test set experimental setups. We used a neural network architecture of three fully connected layers of 32 ReLU activated neurons to allow for sufficient representation capacity to model datasets of various dimensions. ReLU activation is not smooth but achieves near minimax rates for arbitrary smoothness regression functions with a deep architecture (Schmidt-Hieber 2020). This representation space is unnecessarily large for lower-dimensional datasets, so a validation set stopping criterion was used to prevent overfitting. We used $c = 5$ Gaussian mixture components as the output parameter vector of MDNs and WMDNs, to allow for modelling of complex probability densities while remaining relatively low dimensional. We set $d = 1$ as L_1 distance retains meaningfulness in high dimensional space, and set $q = 2$, $b = 20$, $k = 5$ for all experiments as default values. To tune η, MDNs were trained on each dataset with $\eta \in \{10^n | n = -5, -4.5, \ldots, -2\}$. We selected η_{best} as the η which produced the lowest average validation NLL. WMDNs were then trained with $\eta = \eta_{best}, \lambda_u \in \{10^n | n = -2, -1.5, \ldots, 2\}$ to select λ_u. Experiments were run with 50 seeds for each parameter combination. A validation set of size 1,000 was used for early stopping, and up to 10,000 data points were used for the test set. Having a validation or test set larger than the labelled training set is unrealistic but necessary to accurately compare SSL model performance (Oliver et al. 2018).

CDE Experiments

To investigate the performance of our WLR framework in a CDE setting, we perform experiments on on seven regression datasets from the UCI repository (Dua and Graff 2017). Results are shown in Table 1. Results in all tables are displayed as $mean \pm se$, and that any standard errors of 0 are due to rounding. The dimensionality of each dataset is denoted d. For each dataset, we use 3,000 training data points, of which 100, 300 or 1,000 are labelled. We compare performance of WMDN against supervised deep learning CDE algorithms: MDN and the Uncountable Mixture of Asymmetric Laplacians (UMAL) model (Brando et al. 2019). UMAL is a deep learning architecture for CDE, similar to MDNs. However, UMAL avoids the of strong distributional assumption of MDNs by outputting an uncountable mixture of Laplacian distributions and is thus more flexible.

Although NLL is the most natural way to evaluate CDE performance, this metric is difficult to interpret. We additionally measure interpretable metrics by generating predic-

[1] https://github.com/OGraffeuille/Wasserstein-Laplacian-Regularisation

		Negative Log Likelihood			Interval Coverage (%)		Interval Width	
Data	Labels	MDN	UMAL	WMDN	MDN	WMDN	MDN	WMDN
Electric	100	-0.11 ± 0.05	-0.15 ± 0.19	**-0.38 ± 0.04**	85.2 ± 0.8	**90.5 ± 0.5**	**0.47 ± 0.03**	**0.47 ± 0.01**
($d = 6$)	300	**-0.84 ± 0.03**	**-0.79 ± 0.10**	**-0.86 ± 0.02**	89.2 ± 0.5	**94.0 ± 0.3**	**0.32 ± 0.01**	0.38 ± 0.01
	1000	**-1.31 ± 0.02**	-1.16 ± 0.06	-1.19 ± 0.02	90.9 ± 0.4	**95.3 ± 0.3**	**0.26 ± 0.01**	0.33 ± 0.01
Protein	100	1.33 ± 0.01	1.51 ± 0.26	**1.30 ± 0.01**	93.0 ± 0.4	**93.8 ± 0.3**	**2.26 ± 0.03**	**2.32 ± 0.03**
($d = 9$)	300	**1.14 ± 0.01**	1.26 ± 0.04	**1.12 ± 0.01**	92.3 ± 0.2	**93.0 ± 0.2**	**2.12 ± 0.01**	2.18 ± 0.01
	1000	**0.98 ± 0.00**	1.02 ± 0.02	**0.98 ± 0.00**	**93.5 ± 0.1**	**93.5 ± 0.1**	**2.11 ± 0.01**	**2.13 ± 0.01**
Air Quality	100	0.52 ± 0.02	0.50 ± 0.08	**0.40 ± 0.02**	90.1 ± 0.3	**91.6 ± 0.2**	**1.95 ± 0.03**	**1.92 ± 0.03**
($d = 12$)	300	0.22 ± 0.01	0.21 ± 0.02	**0.12 ± 0.01**	91.4 ± 0.2	**92.4 ± 0.2**	1.66 ± 0.03	**1.58 ± 0.02**
	1000	-0.05 ± 0.00	0.02 ± 0.02	**-0.08 ± 0.00**	93.2 ± 0.1	**93.9 ± 0.1**	**1.50 ± 0.01**	**1.50 ± 0.01**
Elevators	100	1.10 ± 0.01	1.42 ± 0.05	**1.05 ± 0.01**	**89.5 ± 0.4**	**89.1 ± 0.4**	**0.61 ± 0.01**	**0.59 ± 0.01**
($d = 18$)	300	0.92 ± 0.01	0.99 ± 0.10	**0.87 ± 0.01**	**91.3 ± 0.2**	**91.2 ± 0.2**	**0.58 ± 0.01**	**0.56 ± 0.01**
	1000	0.68 ± 0.01	0.64 ± 0.03	**0.59 ± 0.01**	**91.9 ± 0.2**	**92.0 ± 0.1**	0.45 ± 0.01	**0.41 ± 0.01**
Parkinsons	100	0.82 ± 0.03	3.05 ± 0.48	**0.59 ± 0.03**	83.9 ± 0.6	**86.9 ± 0.4**	**12.09 ± 0.28**	13.33 ± 0.27
($d = 20$)	300	0.35 ± 0.01	0.53 ± 0.24	**0.13 ± 0.01**	88.6 ± 0.4	**91.8 ± 0.2**	**10.70 ± 0.19**	12.46 ± 0.32
	1000	-0.03 ± 0.01	-0.09 ± 0.06	**-0.27 ± 0.01**	91.7 ± 0.2	**94.3 ± 0.1**	9.63 ± 0.12	**8.92 ± 0.12**
Appliances	100	0.74 ± 0.01	1.02 ± 0.06	**0.57 ± 0.01**	90.5 ± 0.5	**94.9 ± 0.3**	**274.39 ± 9.41**	335.68 ± 8.07
($d = 27$)	300	0.58 ± 0.01	0.76 ± 0.04	**0.47 ± 0.01**	92.3 ± 0.2	**95.9 ± 0.2**	**320.10 ± 5.76**	380.24 ± 5.29
	1000	0.42 ± 0.01	0.52 ± 0.02	**0.35 ± 0.01**	93.5 ± 0.1	**96.1 ± 0.1**	**378.58 ± 3.89**	388.39 ± 4.17
Song Year	100	1.31 ± 0.01	2.05 ± 0.77	**1.24 ± 0.01**	91.5 ± 0.3	**92.8 ± 0.2**	**32.27 ± 0.52**	33.66 ± 0.35
($d = 90$)	300	1.22 ± 0.01	2.40 ± 0.52	**1.15 ± 0.00**	**93.1 ± 0.3**	**93.1 ± 0.2**	**35.09 ± 0.53**	**35.04 ± 0.29**
	1000	1.15 ± 0.00	2.16 ± 0.42	**1.10 ± 0.00**	**94.2 ± 0.2**	**94.5 ± 0.1**	**37.09 ± 0.35**	37.81 ± 0.20

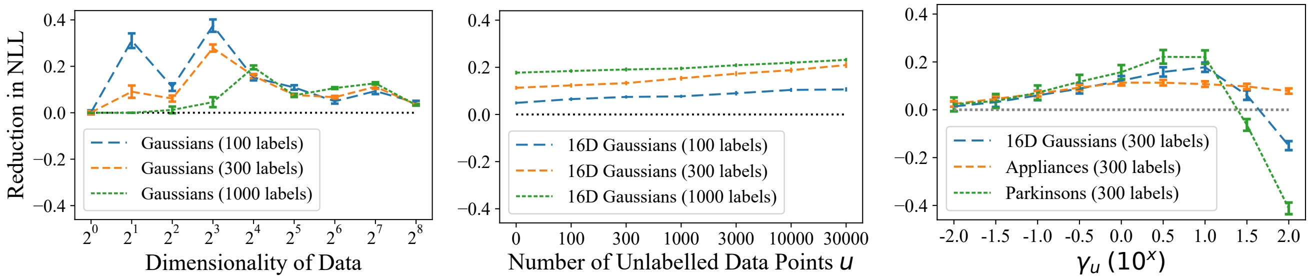
Table 1: Conditional Density Estimation Experiments on UCI Datasets

tion intervals from the models' output densities and computing 1) the percentage of test points that fall within the intervals (Interval Coverage) and 2) the average width of the intervals (Interval Width) (Holmes, Gray, and Isbell 2007). The interval metrics displayed are of 95% confidence.

Performance of WLR Table 1 shows that WMDN outperforms MDN on almost all datasets in terms of NLL. With the Appliances and Song Year datasets, a WMDN with only 100 labelled data points performs equally to a MDN with 300 labelled data points. With the Electric and Parkinsons datasets, a WMDN with only 100 labelled data points performs as similarly to a MDN with 300 labelled data points as to one with 100. For most datasets, WMDN outperforms MDN by a similar amount with 100, 300 or 1,000 labelled data points, indicating that the WLR framework is effective with a range of number of labelled data points. However, WLR does not improve performance of the Electric or Protein datasets with 300 or 1,000 labelled data points. One factor to consider is that these datasets are low dimensional (6D and 9D respectively), hence the underlying data structure is easier to capture with fewer labelled data points. When there are sufficient labelled data points to capture the underlying data structure, unlabelled data cannot improve performance and our WLR framework only adds unnecessary regularisation. In terms of the interval metrics WMDN outperforms MDN in terms of interval cover for almost all datasets, indicating that WLR improved the generalisability of models, but consequentially sometimes performed worse in terms of interval width. UMAL had a higher NLL than MDNs and WMDNs on average, and also higher variance between runs, particularly with few labelled data points. Although the improved flexibility of the UMAL model generally allows it to outperform parametric alternatives in some applications (Brando et al. 2019), it seems to perform worse when few labelled data points are available. UMAL's did not perform as well on these datasets, thus we did not include their interval metric results for this algorithm.

Influence of Data Dimensionality To investigate the effect of dimensionality on the Wasserstein Laplacian Regularisation framework, we generated a series of synthetic datasets. Each dataset is a mixture of five equally weighted multidimensional Gaussians with randomly generated parameters. Datasets of this form were generated in a range of dimensions. The 1D Gaussians dataset is shown in Figure 1. We then compared the performance of MDNs and WMDNs on each dataset; results are shown in Figure 3a.

These results demonstrate two ideas. Firstly, for unlabelled data to improve model performance, the underlying data structure must be complex enough to not be completely captured by the labelled data alone. WLR does not substantially improve performance of low dimensional data as they have a simpler underlying structure, particularly when the model is given more labelled data points. This is consistent with our earlier discussion on the Electric and Protein datasets. Conversely, as the quality of a model produced by a supervised algorithm on a dataset decreases, the gain of performance from unlabelled data also decreases – this is because the unlabelled data will effectively be smoothing an incorrect model (Cozman, Cohen, and Cirelo 2003). Our results show that as the dimensionality of the data increases



(a) versus dimensionality of data

(b) versus quantity of unlabelled data

(c) versus SSL weight coefficient γ_u

Figure 3: Improvement in performance from WLR versus varying parameters.

past an “optimal” dimensionality while the number of available labelled data points remains constant, the quality of a supervised model decreases, and therefore the gain in performance from unlabelled data decreases. Furthermore, the “optimal” dimensionality is higher when more labelled data points are available, as this allows the model to learn a higher quality model in a higher dimensional space.

Influence of Unlabelled Data To investigate the effect varying the quantity of unlabelled data on WLR, we compare the performance of MDNs and WMDNs on datasets with increasing unlabelled data points. Results are shown in Figure 3b. As we increase the quantity of unlabelled data, the gain in performance from the WLR framework increases. This result is consistent with SSL theory (Cozman, Cohen, and Cirelo 2003). An important result is that our WLR framework improves performance with no unlabelled data points. We believe that our framework may therefore be useful be useful as a supervised regularisation framework.

Influence of Parameters The SSL weight coefficient γ_u was tuned independently for each experiment. Figure 3c shows the influence of this parameter on model performance. A small γ_u value will cause little regularisation and thus little improvement, while a large γ_u value will cause WLR to over-regularise and decrease performance. Another parameter of interest is the number of bins b used to approximate $\mathcal{W}_1$ (Equation 8). As discussed in Section , a higher b value increases accuracy of the approximation. Interestingly however, during experiments, we observe that a smaller b actually generates larger gradients with respect to parameters which define f_i, and improves model performance. We recommend $b \in [10, 20]$, values smaller than this will increase the granularity of the distributions computed by $\mathcal{W}_1$ and will thus decrease the ability of WLR to regularise.

Regression Experiments

CDE algorithms are not designed for regression as they optimise for likelihood rather than L_p loss. However, it can sometimes be useful to use CDE models for regression by reducing the estimated density to a single value such as the mean or mode, for example to model inverse problems (Pahlevan et al. 2020).

In order to evaluate the effect of WLR on CDE algorithms in a regression setting, we repeat the experiments from Table 1, but use the distribution mean as our estimate and measure performance with Root Mean Squared Error (RMSE). Additionally, we compare performance to two semi-supervised regression algorithms, Co-Regularisation (CoREG), a co-training kNN algorithm (Zhou and Li 2005), and and Mean Teacher (Tarvainen and Valpola 2017), which we adapted for regression. Mean Teacher is a semi-supervised consistency regularisation-based algorithm which Oliver et al. (2018) found to perform best in a fair evaluation of deep semi-supervised classification algorithms. Results are shown in Table 2.

WLR improved MDN performance for most datasets. The exceptions were the Electric and Protein datasets, consistent with our CDE results. Our WLR framework can thus also improve CDE performance in a regression setting. However, Mean Teacher generally outperformed WMDN. This shows that CDE algorithms are not the best performing approach to modelling regression problems for most datasets.

Chlorophyll-a Estimation Case Study

We evaluated our framework on remote sensing chlorophyll-a datasets; one “Simulated” and one “Real”. For each dataset, the input data is in-situ multispectral reflectance adapted to Sentinel-3 OLCI bands, and the target variable is logged chlorophyll-a concentration (Pahlevan et al. 2020). The Simulated chlorophyll-a dataset was generated by simulating the forward optical problem with random water constituent concentrations. The optical model used is built off (Dekker, Vos, and Peters 2002), but adapted to include chlorophyll-a and a noise component. The Real dataset used is described by Pahlevan et al. (2020). This dataset is the result of international collaboration, and includes 2,612 measurements of chlorophyll-a concentrations and corresponding hyperspectral data, and is not currently publicly available. Parameters used in these experiments are identical to previous sections, with two exceptions: we use $b = 10$ bins to increase performance, and use a 2,000-300-312 for a train-validation-test data split for the Real dataset. Results are shown in Table 3.

In the CDE setting, WLR substantially improved MDN performance for the Simulated dataset, and for the Real dataset with fewer than 1,000 labelled data points. With both datasets, a WMDN with only 100 labelled data points performs as similarly to a MDN with 300 labelled data points

		Root Mean Squared Error				
Data	Labels	MDN	UMAL	CoREG	Mean Teacher	WMDN
Electric $(d = 6)$	100	0.16 ± 0.01	1.24 ± 0.01	0.35 ± 0.00	**0.13 ± 0.01**	**0.15 ± 0.01**
	300	0.11 ± 0.00	1.27 ± 0.01	0.27 ± 0.00	**0.08 ± 0.00**	0.10 ± 0.00
	1000	0.08 ± 0.00	1.32 ± 0.01	0.20 ± 0.00	**0.07 ± 0.00**	0.08 ± 0.00
Protein $(d = 9)$	100	0.71 ± 0.00	0.83 ± 0.01	0.74 ± 0.00	**0.69 ± 0.00**	0.71 ± 0.00
	300	0.68 ± 0.00	0.84 ± 0.00	0.70 ± 0.00	**0.65 ± 0.00**	0.68 ± 0.00
	1000	0.65 ± 0.00	0.86 ± 0.00	0.63 ± 0.00	**0.60 ± 0.00**	0.65 ± 0.00
Air Quality $(d = 12)$	100	0.62 ± 0.01	1.83 ± 0.01	0.77 ± 0.01	**0.56 ± 0.01**	0.61 ± 0.01
	300	0.53 ± 0.00	1.89 ± 0.01	0.66 ± 0.01	**0.49 ± 0.00**	0.51 ± 0.00
	1000	0.46 ± 0.00	1.91 ± 0.01	0.54 ± 0.00	**0.45 ± 0.00**	**0.45 ± 0.00**
Elevators $(d = 18)$	100	0.20 ± 0.00	0.29 ± 0.00	0.20 ± 0.00	**0.14 ± 0.00**	0.19 ± 0.00
	300	0.17 ± 0.00	0.29 ± 0.00	0.18 ± 0.00	**0.11 ± 0.00**	0.16 ± 0.00
	1000	0.13 ± 0.00	0.30 ± 0.00	0.17 ± 0.00	**0.09 ± 0.00**	0.12 ± 0.00
Parkinsons $(d = 20)$	100	4.09 ± 0.08	11.08 ± 0.11	5.47 ± 0.04	**3.55 ± 0.05**	3.98 ± 0.09
	300	3.26 ± 0.04	11.54 ± 0.18	4.31 ± 0.02	**2.74 ± 0.02**	3.05 ± 0.04
	1000	2.76 ± 0.02	12.50 ± 0.25	3.18 ± 0.01	**1.95 ± 0.01**	2.53 ± 0.02
Appliances $(d = 27)$	100	103.69 ± 0.16	110.75 ± 0.23	106.50 ± 1.20	**99.75 ± 0.19**	102.35 ± 0.15
	300	102.57 ± 0.11	110.92 ± 0.15	100.00 ± 0.30	**97.07 ± 0.21**	100.53 ± 0.12
	1000	100.35 ± 0.11	111.24 ± 0.11	96.30 ± 0.20	**93.72 ± 0.24**	98.17 ± 0.16
Song Year $(d = 90)$	100	11.14 ± 0.02	16.53 ± 5.38	11.16 ± 0.05	**10.91 ± 0.05**	11.09 ± 0.02
	300	10.99 ± 0.02	16.88 ± 5.76	10.93 ± 0.02	**10.63 ± 0.03**	10.94 ± 0.01
	1000	10.77 ± 0.02	16.06 ± 4.95	10.68 ± 0.02	**10.21 ± 0.02**	10.73 ± 0.01

Table 2: Regression Experiments on UCI Datasets

		Negative Log Likelihood			Root Mean Squared Error		
Data	Labels	MDN	UMAL	WMDN	MDN	Mean Teacher	WMDN
Simulated $(d = 12)$	100	0.86 ± 0.01	1.04 ± 0.02	**0.79 ± 0.01**	**1.12 ± 0.00**	**1.11 ± 0.01**	**1.12 ± 0.01**
	300	0.71 ± 0.01	0.88 ± 0.04	**0.69 ± 0.00**	1.07 ± 0.00	**1.04 ± 0.00**	1.07 ± 0.00
	1000	0.64 ± 0.00	0.68 ± 0.01	**0.63 ± 0.00**	1.03 ± 0.00	**1.00 ± 0.00**	1.03 ± 0.00
Real $(d = 16)$	100	0.86 ± 0.02	1.42 ± 0.05	**0.68 ± 0.01**	**0.67 ± 0.01**	**0.68 ± 0.02**	**0.65 ± 0.01**
	300	0.53 ± 0.01	0.67 ± 0.06	**0.48 ± 0.01**	0.58 ± 0.01	0.63 ± 0.02	**0.56 ± 0.00**
	1000	**0.22 ± 0.01**	0.26 ± 0.02	**0.22 ± 0.01**	0.49 ± 0.00	0.58 ± 0.01	**0.47 ± 0.00**

Table 3: Conditional Density Estimation and Regression Experiments on Chlorophyll-a Remote Sensing Datasets

as to one with only 100. In the regression setting, both MDN algorithms outperformed Mean Teacher for the Real dataset, showing that when modelling inverse problems with regression, CDE algorithms can outperform regression algorithms. Furthermore, WMDN outperformed MDN in this setting.

In practice, chlorophyll-a models are regional and are trained on local datasets which contain at most a few hundred data points (Syariz et al. 2020; Maier and Keller 2019; Allan et al. 2011). Our results show that for datasets of this size, our framework could allow practitioners to improve estimation performance by as much as if they were to gather substantially more labelled data points.

Conclusions

We proposed Wasserstein Laplacian Regularisation, a framework that allows for semi-supervised learning of Conditional Density Estimation problems. The framework applies a secondary objective function during training of CDE algorithms to ensure that they learn a smooth function along the manifold of underlying data.

We observed that in applications with limited labelled data and abundant unlabelled data, this framework allows CDE algorithms to leverage unlabelled data to substantially improve model performance in terms of NLL. For two of our seven test datasets we find that our framework allows semi-supervised models to achieve equal performance to a supervised model with three times as many labelled data points. Furthermore, we find that in some cases, our framework can improve model performance in a supervised setting when no unlabelled data points is available. We also perform a case study on the application of chlorophyll-a, and find that our framework may substantially improve performance of models used in this field.

Future work could explore alternative approaches to semi-supervised CDE, including applying the WLR framework to different CDE algorithms, or using other SSL techniques such as consistency regularisation.

Acknowledgements

This research is partially funded by the New Zealand MBIE TAIAO data science programme. We thank Nima Pahlevan, Brandon Smith, Ronghua Ma, Natascha Oppelt, John Schalles, Caren Binding, Krista Alikas, Daniela Gurlin, Hà Nguyễn, Bunkei Matsushita, Wes J. Moses and Steven R Greb for the Real chlorophyll-a dataset. We thank Mat Allan for his Simulated chlorophyll-a dataset, and Dennis Wilson for his insight.

References

Allan, M. G.; Hamilton, D. P.; Hicks, B. J.; and Brabyn, L. 2011. Landsat Remote Sensing of Chlorophyll-a Concentrations in Central North Island lakes of New Zealand. *International Journal of Remote Sensing*, 32(7): 2037–2055.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein Generative Adversarial Networks. In *International Conference on Machine Learning*, 214–223. PMLR.

Belkin, M.; Niyogi, P.; and Sindhwani, V. 2006. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *Journal of Machine Learning Research*, 7(85): 2399–2434.

Bishop, C. M. 1994. Mixture Density Networks. Technical report, Aston University.

Bishop, C. M. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag. ISBN 0387310738.

Bisquert, M.; Caselles, E.; Sánchez-Tomás, J.; and Caselles, V. 2012. Application of artificial neural networks and logistic regression to the prediction of forest fire danger in Galicia using MODIS data. *International Journal of Wildland Fire*, 21: 1025–1029.

Brando, A.; Rodríguez-Serrano, J. A.; Vitrià, J.; and Rubio, A. 2019. Modelling heterogeneous distributions with an Uncountable Mixture of Asymmetric Laplacians. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 8836–8846.

Cozman, F. G.; Cohen, I.; and Cirelo, M. C. 2003. Semi-Supervised Learning of Mixture Models. In *Proceedings, Twentieth International Conference on Machine Learning*.

Dekker, A. G.; Vos, R.; and Peters, S. 2002. Analytical algorithms for lake water TSM estimation for retrospective analyses of TM and SPOT sensor data. *International Journal of Remote Sensing*, 23(1): 15–35.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml. University of California, Irvine, School of Information and Computer Sciences. Accessed 2021-06-01.

Holmes, M. P.; Gray, A. G.; and Isbell, C. L. 2007. Fast Nonparametric Conditional Density Estimation. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence*, UAI'07, 175—182. Arlington, Virginia, USA: AUAI Press. ISBN 0974903930.

Huang, B. 2021. Semi-supervised Conditional Density Estimation for Imputation and Classification of Incomplete Instances. *arXiv preprint arXiv:2106.01708*.

Khan, R. R.; and Sugiyama, M. 2012. Least Squares Conditional Density Estimation in semi-supervised learning settings. In *2012 7th International Conference on Electrical and Computer Engineering*, 109–112. IEEE.

Klotz, D.; Kratzert, F.; Gauch, M.; Keefe Sampson, A.; Brandstetter, J.; Klambauer, G.; Hochreiter, S.; and Nearing, G. 2021. Uncertainty Estimation with Deep Learning for Rainfall-Runoff Modelling. *Hydrology and Earth System Sciences Discussions*, 1–32.

Levatić, J.; Ceci, M.; Stepišnik, T.; Džeroski, S.; and Kocev, D. 2020. Semi-supervised regression trees with application to QSAR modelling. *Expert Systems with Applications*, 158: 113569.

Li, Y.-F.; Zha, H.-W.; and Zhou, Z.-H. 2017. Learning safe prediction for semi-supervised regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Luo, M.; Zhang, L.; Nie, F.; Chang, X.; Qian, B.; and Zheng, Q. 2017. Adaptive semi-supervised learning with discriminative least squares regression. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2421–2427.

Maier, P. M.; and Keller, S. 2019. Application of different simulated spectral data and machine learning to estimate the chlorophyll a concentration of several inland waters. In *2019 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, 1–5. IEEE.

Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Oliver, A.; Odena, A.; Raffel, C.; Cubuk, E. D.; and Goodfellow, I. J. 2018. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, 3239—-3250. Red Hook, NY, USA: Curran Associates Inc.

Pahlevan, N.; Smith, B.; Schalles, J.; Binding, C.; Cao, Z.; Ma, R.; Alikas, K.; Kangro, K.; Gurlin, D.; Hà, N.; et al. 2020. Seamless retrievals of chlorophyll-a from Sentinel-2 (MSI) and Sentinel-3 (OLCI) in inland and coastal waters: A machine-learning approach. *Remote Sensing of Environment*, 240: 111604.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc.

Sakai, T. 2018. Comparing two binned probability distributions for information access evaluation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 1073–1076.

Schmidt-Hieber, J. 2020. Nonparametric regression using deep neural networks with ReLU activation function. *The Annals of Statistics*, 48(4): 1875–1897.

Shu, R.; Bui, H. H.; and Ghavamzadeh, M. 2017. Bottleneck conditional density estimation. In *International Conference on Machine Learning*, 3164–3172. PMLR.

Sohn, K.; Lee, H.; and Yan, X. 2015. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28: 3483–3491.

Solomon, J.; Rustamov, R.; Guibas, L.; and Butscher, A. 2014. Wasserstein propagation for semi-supervised learning. In *International Conference on Machine Learning*, 306–314. PMLR.

Sugiyama, M.; Takeuchi, I.; Suzuki, T.; Kanamori, T.; Hachiya, H.; and Okanohara, D. 2010. Least-squares conditional density estimation. *IEICE Transactions on Information and Systems*, 93(3): 583–594.

Syariz, M. A.; Lin, C.-H.; Nguyen, M. V.; Jaelani, L. M.; and Blanco, A. C. 2020. WaterNet: A convolutional neural network for chlorophyll-a concentration retrieval. *Remote Sensing*, 12(12): 1966.

Tarvainen, A.; and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 1195–1204.

Trippe, B. L.; and Turner, R. E. 2018. Conditional density estimation with bayesian normalising flows. *arXiv preprint arXiv:1802.04908*.

Van Oord, A.; Kalchbrenner, N.; and Kavukcuoglu, K. 2016. Pixel recurrent neural networks. In *International Conference on Machine Learning*, 1747–1756. PMLR.

Zhou, Z.-H.; and Li, M. 2005. Semi-Supervised Regression with Co-Training. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, IJCAI'05, 908—-913. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.