Better Modelling Out-of-Distribution Regression on Distributed Acoustic Sensor Data Using Anchored Hidden State Mixup

Hasan Asy’ari Arief, Peter James Thomas, and Tomasz Wiktorski

Abstract—Generalizing the application of machine learning models to situations where the statistical distribution of training and test data are different has been a complex problem. Our contributions in this paper are threefold: (1) we introduce an anchored-based Out of Distribution (OOD) Regression Mixup algorithm, leveraging manifold hidden state mixup and observation similarities to form a novel regularization penalty, (2) we provide a first of its kind, high resolution Distributed Acoustic Sensor (DAS) dataset that is suitable for testing OOD regression modelling, allowing other researchers to benchmark progress in this area, and (3) we demonstrate with an extensive evaluation the generalization performance of the proposed method against existing approaches, then show that our method achieves state-of-the-art performance. Lastly, we also demonstrate a wider applicability of the proposed method by exhibiting improved generalization performances on other types of regression datasets, including Udacity and Rotation-MNIST datasets.

Index Terms—Distributed Acoustic Sensor, Regression Mixup, Out-of-Distribution Regression, DAS dataset.

I. INTRODUCTION

The capability of a machine learning system to accurately model and predict data corresponding to situations with reduced similarity to those covered by the training set is a desirable property and allows for a more reliable and safe deployment in real-world applications. However, deep neural network as the backbone of state-of-the-art machine learning systems often provide incorrect predictions but report falsely high confidence when evaluated on distributional shifts, also called Out-of-Distribution (OOD), dataset [1]. This is problematic because the distribution of the real-world data often covers a much wider range of characteristics compared to those covered by carefully curated training datasets. Therefore erroneous and high confidence predictions are a major roadblock when implementing machine learning techniques in applications sensitive to safety, security and cost.

Collecting more data from diverse scenarios in multi-environment settings is one method to provide robust generalization and can reduce the OOD problem [2]. However in many cases, data collections are expensive and it is often impossible to capture all possible scenarios. For example, in the case of distributed phase-fraction estimation of multiphase fluids, collecting training data from across a wide range of combinations of Water in Liquid Ratio (WLR), Gas Volume Fraction (GVF), fluid flow rate, fluid velocity, pressure and temperature setting, pipe diameter, etc., will lead to an intractable test matrix and experimental design. Fig. 1 depicts a fiber optic installation for collecting the acoustic signature of multiphase fluid flowing in a flow-loop infrastructure. Fig. 2 shows how the fiber optic cable acts as a sensor array for detecting acoustic sources within the surrounding environments. The distributed phase fraction is defined as the percentage (or fraction) of water, oil, and gas in the total mixture of fluids within the pipeline. Distributed phase fraction measurements can provide a game-changing sensing capability in the multi-billion dollar hydrocarbon production industry, and can be deployed as depicted in Fig. 3. The technology also provides environmental benefits such as reducing carbon footprints due to production and reducing the need for new oil field developments while renewable alternatives reach maturity [3].

This work was funded by the Research Council of Norway’s (RCN) Petromaks2 programme (Grant number 308840) and industry partners, namely Equinor and Lundin. The computations and model reproductions were performed on resources provided by UNINETT Sigma2—the National Infrastructure for High Performance Computing and Data Storage in Norway.

(Corresponding author: Hasan Asy’ari Arief.)

H. A. A. and P. J. T. are with NORSCE Norwegian Research Centre AS, 5008 Bergen, Norway (e-mails: hasv@norceresearch.no and peth@norceresearch.no).

T. W. is with the Department of Electrical Engineering and Computer Science, University of Stavanger, 4036 Stavanger, Norway (e-mail: tomasz.wiktorski@uis.no).
A popular and intuitive way to address the generalization problem is by enhancing the regularization capacity of the machine learning system, for example, by using dropout [4], zero-shot learning [5], label smoothing [6], as well as regularization norms, such as Ridge Regularization and Lasso Regularization. In addition, [2] introduced Invariant Risk Minimization (IRM) and [7] proposed Risk Extrapolation (REx) that were trained on multi-environment settings to provide robust generalization outside the training data. Recently, several novel data augmentation strategies have also been proposed, including Input Mixup [8], Manifold Mixup [9], AdaMixup [10], and Output Mixup as an activation function [11]. These papers provided strong experimental evidence for better generalization of the neural network models by interpolating within the training data as well as within the manifold hidden variables.

Building on the interpolation and regularization ideas, this paper proposes a novel technique called OOD Regression Mixup, specifically developed for reducing the distributional shift problem on regression-based datasets. In the regression problem, the distance of the target from two randomly picked data points consists of a range of real values. These values are also the target variables from different data points. The mixup algorithms mostly ignore this underlying condition when interpolating between two data points while performing the augmentation. The proposed OOD Regression Mixup picks up the interpolation idea and builds on the linearity assumption from [8] to proportionally weight the manifold hidden variables of neural network using a contrast sensitive distance kernel from an anchored data point and use this as a regularization signal. We demonstrate that the method provides a strong regularization capacity on regression-based datasets, depicted in Fig. 2. While our work was motivated by modelling the WLR using Distributed Acoustic Sensing (DAS), we found out that our methodology is applicable in other settings, including image regression datasets. The contributions of our work are as follows:

- We introduce a novel anchored-based OOD Regression Mixup algorithm that proportionally weights the manifold hidden variables from a neural network model, leveraging a distance-based kernel and providing a more generalized capability of the machine learning system.
- We provide a unique 450 GB spatio-temporal distributed acoustic sensor dataset from multi-phase fluid flow experiments (also called MPFF-DAS). The dataset covers a wide range of phase-fraction situations and is highly suitable for validating generalization techniques for the complex OOD regression problem.
- We demonstrate our proposal with an extensive evaluation on modelling the OOD datasets, including on the MPFF-DAS, Udacity and rotation-MNIST datasets. We also evaluate several novel generalization techniques on the datasets and show that our method achieves state-of-the-art performance on this challenging problem.

II. BACKGROUND

Suppose we have a training set $D$ of $(x, y)$ pairs sampled from the true distribution $p(x, y) \sim P$, where $x$ is raw input vector and $y$ is a real value ranging between two real values. Modelling the $(x, y)$ pairs is a regression task, with an objective to find a minimum $p(x) - y$, similar to [12]. The OOD setting is considered when the $y$ sampled from $P$ does not belong to $D$, meaning that the $p(x)$ is shifting while $p(y|x)$ is staying the same. In this paper, the focus is mainly on the datasets that have low similarity (high distributional shift) with the test set. Fig. 2 shows the difference of data distribution between the training and test data from the MPFF-DAS dataset, also called the OOD DAS dataset.

Deep neural network algorithms are trained to minimize the average error over the training data using a rule known as Empirical Risk Minimization (ERM) principle [13]. The ERM is used to approximate expected risk by calculating error using the loss function $\ell$ over the true distribution $P$. The approximation is used because the distribution $P$ is usually unknown, especially in complex systems. Therefore, the ERM uses the empirical distribution from the training data to calculate the empirical risk $R_E(f)$, defined as:

$$R_E(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i).$$

However, focusing only on fitting well to the training data and ignoring the potential that training data does not represent $P$. 

![Fig. 2. Illustration of a fiber optic cable as a Distributed Acoustic Sensor. The Interrogation Unit (IU) sends light pulses along the fiber, the backscattering signal travels back to the IU carrying acoustic profile along the cable, depicted from [3].](Image 55x637 to 294x736)

![Fig. 3. Sketch of fiber optic installment for down-hole measurement using Distributed Acoustic Sensor (DAS), depicted from [3].](Image 318x552 to 557x736)
will lead to overfitting and memorization [14]. OOD modelling
techniques, called IRM and Rex, works by using multi-
environment training sets to minimize the variance of ERM
from multi-set training distribution, with aim of achieving
minimum variance on the new unseen distributional shift data
points.

Other generalization techniques build on the assumption
that the training data is not enough to represent the true
distribution of the overall objective, therefore new data or
slight variations of the training data can help overcome this
limitation. Data augmentation strategies build on this idea
by introducing a new representation of the training data. A
classical augmentation technique is implemented by rotating,
colour shifting, flipping, and blurring the input image [15].
It can also work by introducing noise perturbation within the
data [16], or by combining augmentation with random field
algorithm [17]. A more recent data augmentation strategy
is the mixup algorithm [8]. It is based on the Vicinal Risk
Minimization (VRM) principle [18] which uses virtual training
set from vicinity distribution (v) to approximate the true
distribution P. The input mixup method samples from a vicinal
mixup distribution (x, y) ~ v by interpolating pairs of input
data (xi, yi) and (xj, yj) controlled by a random variable α
to produce a new pair of input-output representations (⃗x, ⃗y).
Specifically, the sampling procedure is as follows:

\[ \tilde{x} = \lambda x_i + (1 - \lambda) x_j, \]
\[ \tilde{y} = \lambda y_i + (1 - \lambda) y_j, \]

where \( \lambda \sim \text{Beta}(\alpha, \alpha) \), for \( \alpha \in (0, \infty) \). The \( \alpha \) represent
the strength of interpolations on the input and output pairs.
The other type of mixup is called AdaMixup which learns
the mixing policy regions automatically, and benefits from
mixing multiple inputs for generalization. Lastly, the manifold
mixup focuses on the mixing pairs of hidden variables within
the neural network layers with the aim of improving the
hidden representation and decision boundaries of the neural
network layers. For completeness, as the manifold mixup aims
at modelling the hidden variables, several feature clustering
algorithms can also be considered for optimizing the hidden
variables within the feature spaces. The techniques such as,
rank-constrained clustering algorithm [19], as well as dynamic
affinity graph construction [20], can be considered; the latter


![Fig. 4. Distribution of the test error along the rotation degrees on the Rotation-
MNIST dataset. The grey areas show the existing samples in the training data,
while the white areas show the OOD test data.](image)

![Fig. 5. The MPFF-DAS data distribution for phase-fraction information in
terms of WLR. The x-axis represents the percentage of water in the multiphase
fluids, while the y-axis represents the density distribution from the total data
point.](image)

### III. Regression Model with Mixup

In the next section, we will show some limitations when
using existing mixup algorithms for modelling regression
datasets (Section III-A). We will also show how we overcome
the problem using anchored-based OOD Regression Mixup
(Section III-B), and why using our technique is more desirable
than using the other mixup algorithms for modelling the
regression data (Section III-C).

#### A. Rethinking Mixup

In a regression task, the smaller the distance between
two target values, the more similar these target values are,
unless they represent different objects. In contrast, in a pure
classification task, the class numbers only represent the ID of
each class and have no further meaning. With that in mind,
when sampling from \( v \) in the discretized regression or pure
regression task, the virtual target \( \tilde{y} \) can be an actual target value
lies between \( y_i \) and \( y_k \). Let \( \tilde{y} = y_j \) and \( y_i \leq y_j \leq y_k \), then
the \( y_j \) sampled from training data \( D \) which is a pair of \((x_j, y_j)\),
should have some weight when a model is trained using the
VRM principle to minimize the vicinal risk, as explained in [8].
The input mixup algorithm, however, ignores this
underlying condition by not taking into account the potential
contribution of \( y_j \), when the model was trained on pairs of
\((x_i, y_i)\) and \((x_k, y_k)\). The manifold mixup also ignores (during
training) the empirical risk from \((x_j, y_j)\) when mixing the
hidden states from the aforementioned pairs.

Most of the existing mixup ideas justify the interpolation
approaches by arguing that they can provide better separation
Fig. 6. Schematic of OOD Regression Mixup. $\text{Mix}_\beta(y_i, y_j)$ is explained in Eq. (2) and Eq. (3), respectively. It is worth noting that $\lambda$ for OOD Regression Mixup is not sampled from Beta($\alpha, \alpha$) but it is a fixed number to control the regularization penalty.

between target classes or manifold hidden variables from two different classes. None of them highlighted that the linear interpolation function in the mixup algorithms might work due to the existing linear correlation between two data points. In regression tasks, linear mapping and correlation between two data points exist to some degree and can be quantified by the spatial autocorrelation principle. These similarity and linearity assumptions are based on Tobler’s First Law of Geography: “All things are related, but nearby things are more related than distant things” [21]. Finally, we hypothesize that “the neighbourhood similarities, calculated using distance kernel from $y$, can help better generalize regression model”.

B. Anchored-based OOD Regression Mixup

Directly interpolating input and output pairs in regression datasets depends heavily on the pairs having a strong linear correlation with each other, which is rarely the case in modelling the real-world data. Therefore, rather than focusing on input mixup, the proposed OOD Regression Mixup is implemented in the manifold hidden variables within the neural network. It should be noted that the proposed algorithm does not operate as an augmentation technique, rather it behaves more like an additional cost function mixing the distance between two hidden variables from different data points as a regularization signal, depicted in Fig. 6.

In this paper, the neural network is trained as

$$f(h(x)) = \hat{y},$$

where $h(x)$ is all the layers of neural network before the fully connected layer and $\hat{y}$ denotes the neural network prediction. The output of $h(x)$ is denoted as $z$, and it represents the $n$-dimensional feature vector before the final regression head.

The OOD Regression Mixup minimizes the proportional distance between hidden variables $z_i$ and $z_j$ from two different data points $(x_i, y_i)$ and $(x_j, y_j)$, to enforce the linearity assumption of two similar target variables, $y_i \sim y_j$. The pair $(x_j, y_j)$ is called an anchored data point, because it is used as a pseudo-target variable when calculating the proportional distance between $z_i$ and $z_j$. It must be emphasized that in OOD Regression Mixup, $(x_j, y_j)$ is an actual data point sampled from the training distribution $D$ and trained using the normal ERM principle.

The algorithm is implemented in three steps. First, we calculate the distance between $y_i$ and $y_j$ using a contrast sensitive distance kernel. It is defined as:

$$w_i = exp\left(-\frac{|y_i - y_j|}{\beta^2}\right),$$

where $\beta$ is a fixed number and denotes the limit of the mixup effect. The larger the distance between $y_i$ and $y_j$ the smaller the mixup effect controlled by $\beta$. The distance kernel is a simplification of the smoothness kernel from the Conditional Random Field (CRF) algorithm [22] and the kernel is used because we aim to invoke a similar effect of CRF on measuring contrast smoothness of the targeted (smooth) regression data.

Next, we calculate the proportional distance between $z$ when compared to $y$ denoted as $d$. With $\lambda$ denotes the regularization learning rate controlling the overall effect of the regularization objective, the $d_i$ is defined as:

$$d_i = \lambda \sum_{m=0}^{n} \frac{y_i \times z_{jm} - y_j \times z_{im}}{y_j \times z_{jm}} ,$$

with aim to achieve

$$\frac{y_i}{y_j} \sim \frac{z_i}{z_j}, \text{ when } y_i \sim y_j.$$

It is worth mentioning that the proportional distance is used, instead of other distance metrics e.g. L1 and L2 distances, because it ensures smooth penalty on small-range similar data points while working fairly well on tackling anomalous features on high dimensional feature vector.

Finally, we calculate the OOD Regression Mixup as an additional cost function $(w_i \times |d_i|)$, therefore the neural network minimizes:

$$L(f) = \mathbb{E}_{(x_i, y_i) \sim P} \mathbb{E}_{(x_j, y_j) \sim P} \ell_r(\hat{y}, y_i) + (w_i \times |d_i|).$$

C. Theoretical explanation

The mixup algorithms have shown numerous successes in augmenting the training data to achieve better generalization, not to mention they also improve calibration and predictive uncertainty [23]. The simplistic nature and the minimum memory overhead are additional advantages of this powerful yet robust algorithm. In classification tasks for example, mixing Car and Apple classes could improve the separation of the decision boundary between the two classes. In regression tasks however, that behaviour could have an inverse effect, especially in smooth regression datasets. The decision boundary between the two closest neighbouring targets should be minimized because they might share interdependent similarities.
Let \( z_0 \) and \( z_1 \) be the component of \( z \), in other words
\[
z = \begin{bmatrix} z_0 & \ldots & z_{n-1} & z_n \end{bmatrix};
\]
\( z \) is the discriminative feature vector extracted from the input data of a regression model. Then, there will be \( z_a \in z_i \) and \( z_b \in z_j \) where \( w_a \times w_b \) proportionally comparable to \( z_b \times w_b \) when \( y_i \sim y_j \), otherwise the regression transition is not smooth. \( w_a \) and \( w_b \) denote the fully connected weights in the neural network that maps \( z \) to the prediction \( \hat{y} \). By separating between two target (or classes), the Mixup unfortunately increases the distance between \( z_a \) and \( z_b \), therefore \( z_a \times w_a \neq z_b \times w_b \). The proposed mixup will make sure \( z_a \times w_a \) be more comparable to \( z_b \times w_b \) by targeting the proportional distance of \( y_i/y_j \sim z_i/z_j \) in Eq. \[3\] when minimizing the overall objectives during training.

The \( z_0 \) and \( z_b \) can be thought of as the subset of discriminative features that up to some limit can linearly map the target variables; the mapping limit in OOD Regression Mixup is controlled by \( \beta \). In modelling the WLR from DAS, for example, the Speed of Sound (SoS) can provide smooth linear mapping among the closest phase-fraction values, depicted in Fig. \[8\] Extracting SoS, however, is not trivial and accurate SoS estimation can only be achieved within certain constraints, including having sufficiently high Signal to Noise Ratio (SNR) data covering a sufficient number of spatial channels.

Other examples are found in the Udacity and rotation-MNIST datasets. The Udacity dataset (Udacity 2018) is used for detecting the steering angle of a car from visual representation seen by the driver outside the car. This dataset shows a smooth transition between two closest angles when they are estimated from the same scene and location. The rotation-MNIST dataset, on the other hand, was built by rotating (between 0 and 180 degrees) MNIST images \[24\]; the MNIST dataset was published under CC BY-SA 3.0 license. While it is a toy dataset, it simulates a smooth regression phenomenon. The smaller the distance of the rotation angle between two data points, the more similar the images are, unless the two images represent different objects or class numbers which is common occurrences in multivariate regression. While those examples do not capture all the different behaviours of the regression dataset, we hypothesize that there exists a smooth transition (a.k.a linear correlation) within a small range of neighbouring targets. We quoted from \[8\]: "Linearity is a good inductive bias from the perspective of Occam's razor since it is one of the simplest possible behaviours". Therefore, taking advantage of this condition will provide a better generalization for the given modelling objectives.

As long-range linearity is often not the case for complex real-world data, we argue that the OOD Regression Mixup will be more applicable for OOD regression rather than the normal regression data. In the OOD setting, there are too many unknowns, and when it is OOD, we can safely assume that the data distribution of training and test data is different, hence the term out-of-distribution. Our proposal invokes a common property of smooth regression data: the data with similar target variables should have a similar correlation between features/properties. For example in fluid dynamics, if fluid temperature increases slowly then the fluid volume will also expand slowly. The correlation between temperature and volume in physics is obvious and can be applied to many applications. We argue that bringing this general formulation of linear correlation to the OOD data is more advantageous than methods aiming to fit a model only on training data that has a different distribution compared to the test data. In general regression modelling, the test data is relatively similar to the training data. Therefore, forcing the general formulation of linear correlation using our proposal might be problematic if the training data in actuality cannot be linearly mapped.

Fig. 7. Mixup Regularization training procedure in PyTorch.

Fig. 8. Relation between Speed of Sound and multiphase mixture including WLR and GVF, depicted from \[3\].
The Multiphase Fluid Flow - Distributed Acoustic Sensor (MPFF-DAS) dataset is high-resolution spatio-temporal data, consisting of a high-frequency temporal dimension (40kHz) and high-resolution spatial sampling (0.4 m/sample, 5m gauge length). The dataset contains acoustic information on every location covered by the fiber optic cable. The dataset was obtained from controlled experiments in a flow-loop laboratory with the fiber cables attached in a straight line on the flow-loop pipes. The training and test data were taken at different times with different experimental designs, therefore, they represent the distributional shift behaviour. The training and test data consist of two and four hours of recording, respectively.

The dataset consists of phase-recovered DAS data and log files from the experiments consisting of multi-phase information, including WLR, GVF, fluid flow rate, pressure, temperature, timestamp, and flow velocity. The phase-recovered DAS data with their corresponding labels are in the compressed Numpy format consisting of 30 seconds recording with a median file size of 0.7 GB per file and have been sliced into 5-second intervals with 3-seconds overlap, therefore they are ready to be used for machine learning models. The total size of the MPFF-DAS dataset is around 450 GB and as far as we know, it is the only publicly available DAS dataset relevant to multi-phase fluid flow characterisation.

### B. Generalization on estimating WLR using DAS

In this paper, the MPFF-DAS dataset was used to estimate the distributed phase-fraction in terms of WLR. The distributed phase fraction is defined as the percentage (or fraction) of water, oil, and gas in the total mixture fluids within the oil pipeline; the phase fractions are represented by the WLR and GVF. The WLR, or water cut, represents the volumetric fraction of water within the liquid component of the multiphase fluids and is a key parameter in the context of production optimization in the hydrocarbon industry, see Fig. 9 for block diagram on the use of DAS data for the industry.

Several state-of-the-art deep learning algorithms for DAS data, including ResNet [25], SlowFastNet [26], DETR [27], and Perceiver [28], were used in this experiment. The ResNet model was built using the ResNet model for DAS data with 44-depth building blocks consisting of 42 bottleneck blocks. Each bottleneck block consist of [conv-bn-conv-bn-conv-bn] layers structured sequentially per block. One-third of the bottleneck blocks use stride 1, while the rest use stride 2. The z consist of 256 values, resulted from AdaptiveAvgPool2d of PyTorch. The SlowFastNet on the other hand was trained using transformed input DAS data by reshaping the input data to have four-dimensional data, e.g. sequence, spatial, temporal, and feature dimensions. The fast path of the SlowFastNet architecture samples every 2 items in the temporal dimensions, while the slow path samples every 16 items. Dropout and Batchnorm layers were included in the architecture with the z consisting of 2056 values.

The DETR model was trained using 3 encoders and 1 decoder block, with ResNet50 as the backbone block. 256 generated features are used as the outcome of the transformer layer, where they are being forwarded to the regression head. The last architecture is called Perceiver [28]. The version of Perceiver with a depth of 3 and the number of latent dimensions of 128 was deployed for modelling the DAS data. For the Perceiver, the maximum frequency band was set to 10 and the number of bands was set to 6, following the default setting in the PyTorch version of Perceiver provided in Github[2]. Table I shows the results of modelling the MPFF-DAS dataset using the deep learning algorithms.

The data distribution of the target variables WLR is depicted in Figure 5. Even though the DETR model was originally developed for object detection, it provides the lowest test error among the other models, see Table I. Therefore, we treated the DETR model as the base model to study generalization techniques.

Several generalization techniques were then deployed on the model, including Ridge Regularization, Lasso Regularization, and Perceiver [28].
The MAE is used because it proportionally measures how far off the prediction value is from the actual target value while providing a linear and easy interpretation of the phase-fraction error bars. To avoid statistical errors, we ran each experiment five times and the results are presented in Table II in terms of the mean and standard deviation of MAE from multiple training sessions.

The experimental results show that the OOD Regression Mixup results in smaller absolute errors of 8.28 and 4.49 compared to the baseline ERM model for single and multi-environment settings, respectively. It also validates our hypothesis that Input and Manifold mixup are least suitable for modelling the OOD regression dataset; our proposal provides significant improvement across all the settings in Table II. Interestingly, while the OOD Regression Mixup with IRM provides the lowest error of 26.53 in the multi-environment setting, it is also more stable with a standard deviation of only 0.14. For completeness, we also tested the algorithms by removing time windows in the data corresponding to transitions between different flow conditions when the flow had stopped (both WLR and GVF equal to zero). It shows that our OOD Regression Mixup ($\beta = 1.5$) provides the lowest error of 29.42 ± 0.19, while the baseline ERM model provides an error of 30.95 ± 2.76.

### C. Mixup on other regression datasets

For comparative evaluations, we also tested the proposed OOD Regression Mixup on different datasets, including the Udacity and Rotation-MNIST datasets. The Udacity dataset consists of training and test data taken from videos of several real driving sessions recorded using three different cameras, e.g., left, right, and centre cameras. In this experiment, the training data were acquired only from the centre camera from two different driving scenes, namely HMB_1 and HMB_2. Because we do not have the label for the actual test data, we used the other driving sessions of training data, namely HMB_4, HMB_5, and HMB_6 for the test data and retrieved
Based on all the 1800 target values, then, 5 different target values were removed sequentially for every 5 other different target values (representing a range of 0.5-degree removal per step); the setting is called slice-5. We performed a similar slicing method by using 100 (slice-100) and 500 (slice-500) different target values to represent a more challenging OOD problem representing a range of 10 and 50-degree removal per step, respectively. The test data, on the other hand, was generated using similar process from building the training set, but the whole Rotation-MNIST test data without any removal process (around 18 000 000 data points) were used to evaluate the generalization performances. We ran the experiments five times and reported the mean and standard deviation (std) in terms of MAE for each setting.

The results are presented in Table V and show that the proposal works far superior to other mixup algorithms and is comparable with normal ERM. Subsequently, with a more challenging problem (OOD shift) in slice-500, the proposal works better than ERM, showing that our mixup can work better on the OOD dataset while proposing a good result for the normal regression dataset.

Fig. 4 shows that the input mixup (blue dotted line) works better than our proposal in a small range of rotation degrees but it underfits the data for most of the distribution, making it an unreliable form of OOD generalization technique. On the other hand, our OOD Regression Mixup not only works well along with the training distribution and performs better than the empirical model within this distribution, but it also works superior on the OOD data where the empirical data are non-existent. These results confirm our hypothesis further, that by leveraging the linearity of two similar data points and their manifold hidden variables, our proposal can provide a model that is not only fitting well with the existing data but can only generalize on the unseen OOD data; a characteristic that does not exist in the ERM model and ignored by the existing mixup-based techniques in the benchmark.

D. Computation cost

The computation cost from our proposal comes from calculating the distance kernel (Eq. 2) and proportional distance of $z$ (Eq. 3); they have a linear complexity implemented using simple mathematical formulations. For example, on modelling the MPFF-DAS dataset on inference/test mode, 11.64 and 10.93 samples per second can be proceeded by the ERM and OOD Regression Mixup models, respectively. It is worth noting that the network speed, hard-drive utilization, and CPU load are varied during the processing time, thus making the small difference in inference time between the two models can be ignored.

In this study, each sample contains 5 seconds of information, thus processing a few samples per second for the MPFF objective can provide a real-time application. Moreover, as the additional computation cost from our algorithm is negligible on modern hardware, the proposal can be implemented to better regularize the existing real-time deep learning algorithms, including Fast RCNN, YOLO, and EfficientNet on embedded devices.
TABLE V

| \( \beta \backslash \lambda \) | 1e-05  | 1e-04  | 0.001 | 0.01  | 0.10 | 1.00 |
|--------------------------|-------|-------|-------|-------|------|------|
| 0.10                     | 17.96 | 18.07 | 18.01 | 17.66 | 17.79 | 22.77 |
| 0.50                     | 17.36 | 17.33 | 17.50 | 17.42 | 19.52 | 20.05 |
| 1.10                     | 16.84 | 17.24 | 17.54 | 17.19 | 17.08 | 19.84 |
| 1.50                     | 17.43 | 17.83 | 17.37 | 17.01 | 17.81 | 19.81 |
| 3.00                     | 17.84 | 18.44 | 17.50 | 16.88 | 18.11 | 22.85 |
| 5.00                     | 17.49 | 17.69 | 17.34 | 16.75 | 23.62 | 34.92 |
| 9.00                     | 16.99 | 17.22 | 18.04 | 21.18 | 25.03 | 31.56 |
| 20.00                    | 16.95 | 17.63 | 18.77 | 19.76 | 25.96 | 49.05 |

E. Ablation study

We turn to the ablation study to see the effect of each parameter of our proposal on the OOD setting of the rotation-MNIST dataset. The proposal uses \( \beta \) and \( \lambda \) as additional variables, therefore, we used several different \( \beta \) and \( \lambda \) on the ablation experiments, and the results are presented in Table V. The results show that by providing a relatively low \( \lambda \), the value of \( \beta \) can relatively be ignored on achieving a lower regression error. The same results can also be achieved by providing a low \( \beta \) and ignoring the values of \( \lambda \), depicted in Fig. 10a and Fig. 10b. These interesting results simplify the search space for finding the optimal values for both \( \beta \) and \( \lambda \). By keeping the low values of \( \lambda \), therefore \( \lambda \leq 1 \times 10^{-4} \) and \( \beta \sim 1.0 \), the proposal could achieve a good generalization performance, as shown empirically in Table V and Fig. 10.

The ablation study also reveals that setting up the \( \lambda \) values high makes the generalization results unreliable. This phenomenon is expected, because a high \( \lambda \) enforces the linearization within the feature spaces towards rank-1, limiting the capability of the underlying model to represent the complexity of the input data. To secure the solution space, we suggest the \( \lambda \) is set between 0.01 and \( 1 \times 10^{-6} \), we argue that setting \( \lambda \) lower than \( 1 \times 10^{-6} \) will make the model perform very similar to ERM because a very low \( \lambda \) will make the Mixup Regularization Loss (in Eq. 4) equals to the ERM Loss (Eq. 1).

F. Limitations

The main limitation of our work is on choosing the optimized values of \( \beta \) and \( \lambda \) because we do not have a good theoretical basis and optimization strategy to find the optimized version for both parameters. However, based on the experimental study, setting \( \beta = 1.1 \) and \( \lambda = 1 \times 10^{-4} \) works fine for many cases. Lastly, because we aim to minimize the proportional distances on \( z \) while considering similarity values on \( y \), the algorithm might only work on smooth regression datasets and might not work on high volatility regression data, such as the stock market datasets. This limitation, unfortunately, makes the proposal have limited use outside the smooth regression datasets.

The future studies of our work include: firstly, to find a way to automate the finding of the optimum values of the parameters in our algorithm, therefore the algorithm can be parameter-free and is easier to be adopted within the community. The ablation study shows a possible correlation between \( \lambda \) and \( \beta \), which suggest that optimization techniques, such as gradient-based search and genetic algorithms, can be used for optimizing the two dependent variables. Secondly, as our research aims to optimize hydrocarbon production, the future works must emphasize the use of the MPFF-DAS dataset. This dataset is unique and can help provide a game-changing functionality for monitoring hydrocarbon production in the oil and gas industries. Moreover, as the research in acoustic signal progress, our proposal can also be a complement to other research in the field, including fault diagnostic with machine learning. Acoustic signals in low frequencies for fault diagnostic have intrinsic linearization features, e.g., smoother changes along the time dimension during the acoustic events. We argue that this characteristic make the proposed generalization algorithm more compatible with the research objectives.

V. Concluding Remarks

We have presented the OOD Regression Mixup, a simple and intuitive cost function based on the properties of existing mixup algorithms and regularization norms to provide a better way to model regression data, especially for OOD datasets. We also have provided a spatio-temporal dataset, MPFF-DAS. This dataset is unique and provides a stepping stone to understand the wide range of acoustic signatures from multiphase fluids captured by the fiber optic cable.

Throughout an extensive evaluation, we have shown that our OOD Regression Mixup algorithm provides a much lower regression error compared to the existing ERM models and other mixup algorithms on several regression-based datasets, including Udacity, Rotation-MNIST, and the spatio-temporal MPFF-DAS dataset. In our experiments, we found out that the OOD Regression Mixup works well with several different types of neural network architectures, including the ResNet, 3D CNN, and attention based-model. The algorithm also provides a robust generalization model as indicated by small training session stochasticity based on the values of standard deviations from multiple experiments. Moreover, using a relatively small \( \lambda \), we have shown that the experimental evidence supports our hypothesis that some degree of linearity exists within the regression data, and taking advantage of this could provide a better generalization model for the corresponding dataset, especially for the OOD datasets.

REFERENCES

[1] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayan, “Likelihood ratios for out-of-distribution detection,” in Advances in Neural Information Processing Systems, vol. 32, 2019.
[2] M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz, “Invariant risk minimization,” arXiv preprint arXiv:1907.02893, 2019.
[3] H. A. Arief, T. Wiktorski, and P. J. Thomas, “A survey on distributed fibre optic sensor data modelling techniques and machine learning algorithms for multiphase fluid flow estimation,” Sensors, vol. 21, no. 8, p. 2801, 2021.
[4] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” The journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.
