Denoising Mixup for Regression

Zhengzhang Hou[1, 2], Zhanshan Li[1, 2], Yanbo Liu[4], Geoff S. Nitschke[5], You Lu[6], Ximing Li[1, 2, 3]*

[1]College of Computer Science and Technology, Jilin University, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of the MoE, Jilin University, China
[3]RIKEN Center for Advanced Intelligence Project, Japan
[4]Department of Computer Science, University of Pretoria, South Africa
[5]Department of Computer Science, University of Cape Town, South Africa
[6]Artificial Intelligence Research Institute, Shenzhen University of Advanced Technology, China
{houzhengzhang99, jesse.yanbo.liu, youlu1206, liximing86}@gmail.com, lizs@jlu.edu.cn, gnitschke@cs.uct.ac.za

Abstract

Data augmentation is an intuitive solution to increase the diversity of training instances in the machine learning community. Mixup is acknowledged as an effective and efficient mix-based data augmentation method, following a linear alignment assumption that the linear interpolations of features align the corresponding linear interpolations of labels. Unfortunately, this assumption can be violated in many complex scenarios, resulting in augmented instances with noisy labels, especially for regression problems. To solve this problem, we propose an easy-to-implement mixup method, namely **DE**nosing **MIXUP** (**DE-MIXUP**), which iteratively corrects the noisy response targets by leveraging an auxiliary noise estimation task with mixup deep features. Additionally, we suggest an efficient optimization method with alternating direction method of multipliers. We compare DE-MIXUP with the existing mixup variants and other prevalent data augmentation methods across benchmark regression datasets. Empirical results indicate the effectiveness of DE-MIXUP under the in-distribution and out-of-distribution cases.

Introduction

Modern machine learning algorithms often require abundant training instances to maintain high performance. However, acquiring and labeling instances is intractable and labor-intensive in many real-world scenarios, resulting in scarce data. An intuitive solution is data augmentation, which refers to strategies to increase the diversity of training instances, rather than acquiring more real instances. Data augmentation has been proven effective and efficient (Zha et al. 2024), and many data augmentation strategies have recently been proposed to handle various data modalities such as images (Xu et al. 2023) and texts (Feng et al. 2024).

Among existing strategies, mixup is a modality-independent and, more significantly, straightforward-yet-effective mix-based data augmentation method (Zhang et al. 2018; Cao et al. 2024). Formally, mixup follows a **linear alignment assumption** that the linear interpolations of features align the corresponding linear interpolations of labels; accordingly, it generates augmented instances by linearly interpolating any two labeled instances. During the past decade, many variants of mixup have been investigated to adapt various scenarios (Guo, Mao, and Zhang 2019; Venkataramanan et al. 2020; Hwang and Whang 2021; Yao et al. 2022; Greenewald et al. 2021; Bouniot, Mozharovskyi, and d'Alché-Buc 2024). For example, manifold mixup directly interpolates hidden states to regularize augmented data (Verma et al. 2019); k-mixup extends mixup by using multiple (k) points instead of only pairs of points (Greenewald et al. 2021); local mixup proposed a loss function with weights based on the distance between pairs of mixed samples, effectively reducing the impact of augmented instances that are out-of-distribution (Baena, Drumetz, and Gripon 2022). Although the simplicity of mixup and its variants, they have been successfully applied to a wider range of applications and achieved promising performance improvements (Guo 2020; Franchi et al. 2021; Han et al. 2022).

Despite the simplicity and effectiveness of mixup, its basic linear alignment assumption can be violated in many complex scenarios (Kou et al. 2025a), resulting in augmented instances with noisy labels (Liu et al. 2023). Especially in regression problems whose labels are continuous response targets, the noisy problem of mixup is ubiquitous. For example, in image age estimation tasks, people of the same age correspond to completely different images, where the linear alignment assumption is clearly violated. Previous studies have shown that this noisy problem of mixup degrades regression performance and hurts the generalization ability, even resulting in a U-shaped generalization curve (Yao et al. 2022; Liu et al. 2023).

To solve the noisy problem of mixup in regression, we propose an easy-to-implement method, namely **DE**nosing **MIXUP** (**DE-MIXUP**). We take inspiration from the empirical observations from both previous studies (Zhang et al. 2018; Verma et al. 2019; Baena, Drumetz, and Gripon 2022; Yao et al. 2022) and our early experimental results, where manifold mixup is superior to mixup for regression in most cases. These indirectly imply that the noisy problem in the deep feature space can be less significant than the one in the original feature space. Upon this observation, we propose to refine the noisy response targets caused by mixup with aug-

*Corresponding author

mented deep features. Specifically, we incorporate a noise estimation layer to estimate the noise of mixup response targets by leveraging the mixup deep features. We then fit the regressor by using the refined response targets. Inspired by (Gillberg et al. 2016), we treat the refined response targets and noises as trainable variables and propose an efficient training method by applying the alternating direction method of multipliers. We conduct extensive experiments to compare DE-MIXUP with the existing mixup variants and other prevalent data augmentation methods across benchmark regression datasets. Empirical results indicate the effectiveness of DE-MIXUP under the in-distribution and out-of-distribution cases.

In a nutshell, the contributions of this paper are listed as follows.

- We propose an easy-to-implement mixup-based method for regression named DE-MIXUP with a denoising strategy.
- We suggest an efficient training method by applying the alternating direction method of multipliers.
- We conduct extensive experiments to validate the effectiveness of DE-MIXUP under the in-distribution and out-of-distribution cases.

Related Works

Mixup Augmentation

Mixup (Zhang et al. 2018) is a simple yet effective data augmentation method that has inspired numerous subsequent studies. For example, manifold mixup (Verma et al. 2019) performs linear interpolation on the trainable deep features in the hidden space; mixupE(Zou et al. 2023) approximates the dominant term using results during forward propagation; k-mixup (Greenewald et al. 2021) generates more augmented instances by perturbing two instance groups and interpolating them using the Wasserstein distance; remix (Chou et al. 2021) separates interpolation into label space and input space. Additionally, cutmix (Yun et al. 2019) and its variants (Venkataramanan et al. 2020; Hong, Choi, and Kim 2021; Baek, Bang, and Shim 2021) perform mixup by using nonlinear interpolation on images through cutting and pasting patches; saliency-based methods (Kim, Choo, and Song 2020; Uddin et al. 2021) further extract saliency features to select meaningful pairs of mixed images.

Although mixup has been effectively applied in numerous practical fields, adamixup (Guo, Mao, and Zhang 2019) highlights considerable noise resulting from conflicts between the labels of augmented and original instances. Similarly, sk-mixup (Bouniot, Mozharovskyi, and d'Alché-Buc 2024) argues that the likelihood of label noise increases with the distance between the mixed data. Therefore, local mixup (Baena, Drumetz, and Gripon 2022) suggests reducing the weight of distant input samples to alleviate label noise. Notably, c-mixup (Yao et al. 2022) proposes a more generalized method using a Gaussian kernel to achieve selective interpolation. Furthermore, metamixup (Mai et al. 2022) also introduce meta-learning techniques to provide cleaner instances. In contrast, DE-MIXUP adopts a post-processing strategy to eliminate noise from augmented labels in regression.

Regression faces more significant challenges compared to classification when using mixup, due to the continuous label space, which can result in arbitrarily-incorrect labels. Regmix (Hwang and Whang 2021) utilizes reinforcement learning to identify the most optimal neighboring samples for mixup. Meanwhile, c-mixup (Yao et al. 2022) adjusts the probabilities of mixup based on label similarity to reduce noise in regression tasks. Additionally, ada (Schneider, Goshtasbpour, and Perez-Cruz 2023) and rc-mixup (Hwang, Kim, and Whang 2024) are variants of c-mixup; the former enables interpolation based on cluster membership to provide more training examples, while the latter introduces multi-round robust training to preserve clean label patterns. Notably, since label noise in regression is arbitrary, DE-MIXUP incorporates a noise estimation layer to accurately get the noise value of the augmented labels, thereby extending the mixup effectively to regression tasks.

Deep Regression

Deep learning has been extensively adopted for extracting deep features, leading to more robust regression predictions. Deep regression effectively addresses real-world challenges across various fields, including finance (Zhang, Aggarwal, and Qi 2017), healthcare (de Vente et al. 2020), and physics (Sial et al. 2020). However, real-world regression tasks are often affected by noisy response targets, posing significant challenges to the performance. To address this, the superloss (Castells, Weinzaepfel, and Revaud 2020) is designed to automatically downweight the contribution of noisy "hard samples". Especially for ordinal regression problems, a novel method proposed by (Franchi et al. 2021) handles class-conditional label noise. ConFrag (Kim et al. 2024) further improves the selection of clean samples by training more discriminative representations (**disjoint yet contrastive fragments**). DE-MIXUP focuses on the specific label noise in regression tasks caused by mixup and aims to provide a framework for promoting the adoption of more effective data augmentation techniques in regression tasks.

DE-MIXUP for Regression

In this section, we introduce the proposed **DE**nosing **MIXUP** (**DE-MIXUP**) method for regression problems.

Preliminaries

Notation of regression Let $\mathbf{x} \in \mathbb{R}^d$ and $y \in \mathbb{R}$ denote a d-dimensional feature vector and a response target in regression problems, respectively. Given a collection of n labeled instances $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, the goal of regression is to train a regressor that can predict the response target for any future instance. Generally, the regressor is composed of a deep feature encoder $h_{\mathbf{\Phi}}$ parameterized by $\mathbf{\Phi}$ and a regression layer $f_{\mathbf{W}}$ parameterized by $\mathbf{W}$. The deep feature encoder is used to transform any original feature to a more discriminative deep feature $\mathbf{z}^{\mathbf{\Phi}} = h_{\mathbf{\Phi}}(\mathbf{x})$, where $\mathbf{z}^{\mathbf{\Phi}}$ denotes a trainable deep feature with respect to $\mathbf{\Phi}$. The regression layer is used to generate the prediction of response target $f_{\mathbf{W}}(\mathbf{z}^{\mathbf{\Phi}})$.

Formally, the generic objective of regression problems is given as follows:

$$\mathcal{L}(\mathbf{\Phi}, \mathbf{W}) = \sum_{i=1}^{n} \ell \left(f_{\mathbf{W}} \left(h_{\mathbf{\Phi}} \left(\mathbf{x}_i \right) \right), y_i \right), \tag{1}$$

where ℓ can be any commonly used loss function for regression problems.

Mixup It is a prevalent data augmentation method that linearly combines any instance pair to generate augmented instances (Zhang et al. 2018). Typically, it randomly draws two labeled instances $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$ and then combine them to generate an augmented instance $(\overline{\mathbf{x}}_{ij}, \overline{y}_{ij})$ as follows:

$$\overline{\mathbf{x}}_{ij} = \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j, \quad \overline{y}_{ij} = \lambda y_i + (1-\lambda) y_j \\ \lambda \sim \mathrm{Beta}(\gamma, \gamma), \tag{2}$$

where $\mathrm{Beta}(\gamma, \gamma)$ denotes a pre-defined Beta distribution used to instance the coefficient weight λ.

Beyond the standard version, manifold mixup (Verma et al. 2019) combines the trainable deep features rather than the original features. For any labeled instance pair $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_j, y_j)$, after transforming $\mathbf{x}_i, \mathbf{x}_j$ to $\mathbf{z}_i^{\mathbf{\Phi}}, \mathbf{z}_j^{\mathbf{\Phi}}$, it generates a latent trainable augmented instance $(\overline{\mathbf{z}}_{ij}^{\mathbf{\Phi}}, \overline{y}_{ij})$ as follows:

$$\overline{\mathbf{z}}_{ij}^{\mathbf{\Phi}} = \lambda \mathbf{z}_i^{\mathbf{\Phi}} + (1-\lambda)\mathbf{z}_j^{\mathbf{\Phi}}, \quad \overline{y}_{ij} = \lambda y_i + (1-\lambda) y_j \\ \lambda \sim \mathrm{Beta}(\gamma, \gamma), \tag{3}$$

Regression with mixup We briefly describe the stochastic optimization process of regression with augmented instances of mixup. At each iteration t, it randomly draws a mini-batch $\Omega^{(t)}$ of n_b labeled instances and then generates $n_b(n_b-1)$ augmented instances of mixup. Accordingly, the stochastic objective can be formulated as follows:

$$\mathcal{L}_{\mathbf{m}}(\mathbf{\Phi}, \mathbf{W}) = \sum_{i,j \in \Omega^{(t)}} \ell \left(f_{\mathbf{W}} \left(h_{\mathbf{\Phi}} \left(\overline{\mathbf{x}}_{ij} \right) \right), \overline{y}_{ij} \right) \tag{4}$$

By analogy, the stochastic objective with augmented instances of manifold mixup can be formulated as follows:

$$\mathcal{L}_{\mathbf{M}}(\mathbf{\Phi}, \mathbf{W}) = \sum_{i,j \in \Omega^{(t)}} \ell \left(f_{\mathbf{W}} \left(\overline{\mathbf{z}}_{ij}^{\mathbf{\Phi}} \right), \overline{y}_{ij} \right) \tag{5}$$

DE-MIXUP

Recalling Eqs.(3) and (4), we note that the family of mixup follows a linear alignment assumption between (deep) features and response targets. That is, it supposes that the real regressor is a linear mapping function. Unfortunately, this assumption can be violated in many complex scenarios (Kou et al. 2025b), so the response targets of augmented instances of mixup are inevitably noisy to some extent. Several previous studies (Yao et al. 2022; Liu et al. 2023) have attracted this noisy problem, and they have shown an interesting empirical phenomenon that mixup and manifold mixup degrade regression performance while manifold mixup is superior to mixup in most cases; in our experiments, we have observed similar results (see more results in **Sections 4.2, 4.3**). All these empirical observations indirectly indicate the existence of this noisy problem and, more significantly, they imply that the noisy problem in manifold mixup can be less significant than the one in mixup.

We take inspiration from this observation and refine the noisy response targets caused by mixup with augmented deep features. Specifically, we incorporate a noise estimation layer $g_{\mathbf{B}}$ parameterized by $\mathbf{B}$, which is used to estimate the noise of mixup response targets Δ_{ij} by leveraging the mixup deep features $\overline{\mathbf{z}}_{ij}^{\mathbf{\Phi}}$. So the regression layer can be fitted by using the refined response targets $y_{ij} = \overline{y}_{ij} - \Delta_{ij}$. Inspired by (Gillberg et al. 2016), we treat all y_{ij} and Δ_{ij} as trainable variables and, referring to Eq.(4), we formulate the following stochastic objective l_{DE} of DE-MIXUP:

$$\begin{aligned} \min_{\mathbf{\Phi}, \mathbf{W}, \mathbf{B}, \mathbf{y}^{(t)}, \mathbf{\Delta}^{(\mathbf{t})}} \quad & \sum_{i,j \in \Omega^{(t)}} \Big(\| f_{\mathbf{W}} \left(h_{\mathbf{\Phi}} \left(\overline{\mathbf{x}}_{ij} \right) \right) - y_{ij} \|_2^2 \\ & + \| g_{\mathbf{B}} \left(\overline{\mathbf{z}}_{ij}^{\mathbf{\Phi}} \right) - \Delta_{ij} \|_2^2 \Big) + \alpha \| \mathbf{\Delta}^{(t)} \|_2^2 \\ \textbf{s.t.} \quad & \overline{\mathbf{y}}^{(t)} = \mathbf{y}^{(t)} + \mathbf{\Delta}^{(t)}, \end{aligned} \tag{6}$$

where $\overline{\mathbf{y}}^{(t)}$, $\mathbf{y}^{(t)}$ and $\mathbf{\Delta}^{(t)}$ denote the vector forms of all $\overline{y}_{ij}, y_{ij}$ and Δ_{ij} corresponding to augmented instances from $\Omega^{(t)}$, respectively; $\| \cdot \|_2$ is ℓ_2 norm; and α is the regularization coefficient. To avoid trivial solutions, we employ the ℓ_2 norm to constrain $\mathbf{\Delta}^{(t)}$.

By analogy, the stochastic objective of DE-MIXUP with augmented instances of manifold mixup, dubbed **DE-MMIXUP**, can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{\Phi}, \mathbf{W}, \mathbf{B}, \mathbf{y}^{(t)}, \mathbf{\Delta}^{(\mathbf{t})}} \quad & \sum_{i,j \in \Omega^{(t)}} \Big(\| f_{\mathbf{W}} \left(\overline{\mathbf{z}}_{ij}^{\mathbf{\Phi}} \right) - y_{ij} \|_2^2 \\ & + \| g_{\mathbf{B}} \left(\overline{\mathbf{z}}_{ij}^{\mathbf{\Phi}} \right) - \Delta_{ij} \|_2^2 \Big) + \alpha \| \mathbf{\Delta}^{(t)} \|_2^2 \\ \textbf{s.t.} \quad & \overline{\mathbf{y}}^{(t)} = \mathbf{y}^{(t)} + \mathbf{\Delta}^{(t)} \end{aligned} \tag{7}$$

Training For simplicity, we only introduce the training process of Eq.(6) but omit the one of Eq.(7) because their training processes are almost the same. As directly solving Eq.(6) is intractable, we apply the alternating direction method of multipliers (Boyd et al. 2011), and convert Eq.(6) into an augmented Lagrange problem with Lagrange parameter $\mathbf{\Theta}$ as follows:

$$\begin{aligned} \min_{\mathbf{\Phi}, \mathbf{W}, \mathbf{B}, \mathbf{y}^{(t)}, \mathbf{\Delta}^{(\mathbf{t})}, \mathbf{\Theta}} \quad & \sum_{i,j \in \Omega^{(t)}} \Big(\| f_{\mathbf{W}} \left(h_{\mathbf{\Phi}} \left(\overline{\mathbf{x}}_{ij} \right) \right) - y_{ij} \|_2^2 \\ & + \| g_{\mathbf{B}} \left(\overline{\mathbf{z}}_{ij}^{\mathbf{\Phi}} \right) - \Delta_{ij} \|_2^2 \Big) + \alpha \| \mathbf{\Delta}^{(t)} \|_2^2 \\ & + \frac{\tau}{2} \| \overline{\mathbf{y}}^{(t)} - \mathbf{y}^{(t)} - \mathbf{\Delta}^{(t)} + \frac{\mathbf{\Theta}}{\tau} \|_2^2, \end{aligned} \tag{8}$$

where τ is the penalty parameter. we then optimize the variables of interest $\{\mathbf{\Phi}, \mathbf{W}, \mathbf{B}, \mathbf{y}^{(t)}, \mathbf{\Delta}^{(\mathbf{t})}, \mathbf{\Theta}\}$ by an alternating fashion.

[**Update** $\{\mathbf{\Phi}, \mathbf{W}, \mathbf{B}\}$] When $\{\mathbf{y}^{(t)}, \mathbf{\Delta}^{(\mathbf{t})}, \mathbf{\Theta}\}$ are fixed, the corresponding sub-objective can be reformulated as follows:

$$\begin{aligned} \min_{\mathbf{\Phi}, \mathbf{W}, \mathbf{B}} \sum_{i,j \in \Omega^{(t)}} & \Big(\| f_{\mathbf{W}} \left(h_{\mathbf{\Phi}} \left(\overline{\mathbf{x}}_{ij} \right) \right) - y_{ij} \|_2^2 \\ & + \| g_{\mathbf{B}} \left(\overline{\mathbf{z}}_{ij}^{\mathbf{\Phi}} \right) - \Delta_{ij} \|_2^2 \Big) \end{aligned} \tag{9}$$

we can update them directly using the stochastic gradient method.

[**Update** $\mathbf{y}^{(t)}$] When $\{\mathbf{\Phi}, \mathbf{W}, \mathbf{B}, \mathbf{\Delta^{(t)}}, \mathbf{\Theta}\}$ are fixed, the sub-objective with respect to $\mathbf{y}^{(t)}$ can be reformulated as follows:

$$\min_{\mathbf{y}^{(t)}} \sum_{i,j\in\Omega^{(t)}} \|f_{\mathbf{W}}\left(h_{\mathbf{\Phi}}\left(\overline{\mathbf{x}}_{ij}\right)\right) - y_{ij}\|_2^2 + \frac{\tau}{2}\|\overline{\mathbf{y}}^{(t)} - \mathbf{y}^{(t)} - \mathbf{\Delta}^{(t)} + \frac{\mathbf{\Theta}}{\tau}\|_2^2, \tag{10}$$

This is a convex optimization and its closed solution is given below:

$$\mathbf{y}^{(t)} = (2+\tau)^{-1}(2\mathbf{p}^{(t)} + \tau\overline{\mathbf{y}}^{(t)} - \tau\mathbf{\Delta}^{(t)} + \mathbf{\Theta}) \tag{11}$$

where $\mathbf{p}^{(t)}$ denotes the vector form of all $p_{ij} = f_{\mathbf{W}}\left(h_{\mathbf{\Phi}}\left(\overline{\mathbf{x}}_{ij}\right)\right)$.

[**Update** $\mathbf{\Delta^{(t)}}$] When $\{\mathbf{\Phi}, \mathbf{W}, \mathbf{B}, \mathbf{y}^{(t)}, \mathbf{\Theta}\}$ are fixed, the sub-objective with respect to $\mathbf{\Delta^{(t)}}$ can be reformulated as follows:

$$\min_{\mathbf{\Delta^{(t)}}} \sum_{i,j\in\Omega^{(t)}} \|g_{\mathbf{B}}\left(\overline{\mathbf{z}}_{ij}^{\mathbf{\Phi}}\right) - \Delta_{ij}\|_2^2 + \alpha\|\mathbf{\Delta}^{(t)}\|_2^2 + \frac{\tau}{2}\|\overline{\mathbf{y}}^{(t)} - \mathbf{y}^{(t)} - \mathbf{\Delta}^{(t)} + \frac{\mathbf{\Theta}}{\tau}\|_2^2, \tag{12}$$

This is a convex optimization and its closed solution is given below:

$$\mathbf{\Delta^{(t)}} = (2+\tau+2\alpha)^{-1}(2\mathbf{q}^{(t)} + \tau\overline{\mathbf{y}}^{(t)} - \tau\mathbf{y}^{(t)} + \mathbf{\Theta}) \tag{13}$$

where $\mathbf{q}^{(t)}$ denotes the vector form of all $q_{ij} = g_{\mathbf{B}}\left(\overline{\mathbf{z}}_{ij}^{\mathbf{\Phi}}\right)$.

[**Update** $\mathbf{\Theta}$] It can be directly updated by the following formula:

$$\mathbf{\Theta} \leftarrow \mathbf{\Theta} + \tau(\mathbf{y}^{(t)} + \mathbf{\Delta^{(t)}} - \overline{\mathbf{y}}^{(t)}) \tag{14}$$

Implementation We now introduce some training details. First, referring to Eqs.(11) and (12), $\mathbf{y}^{(t)}$ and $\mathbf{\Delta}^{(t)}$ are estimated by predictions. To avoid inaccurate estimates in the early training stages, we perform a warm-up stage with the original labeled dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$. Then, for each mini-batch, we perform an inner loop to estimate $\mathbf{y}^{(t)}$ and $\mathbf{\Delta}^{(t)}$. For clarity, we summarize the training process of DE-MIXUP in *Algorithm 1*.

Theoretical Analysis

Potential Noisy Pattern

Firstly, we find that there must be a certain noisy pattern between augmented samples and the labels. Rewriting the constraints related to the noise as follows:

$$\mathbf{\Delta}_{ij} = \overline{\mathbf{y}}_{ij} - \mathbf{y}_{ij}^* \tag{15}$$

where $\mathbf{y}_{ij}^* = f^*(\overline{x}_{ij})$, and f^* represents the true regressor corresponding to the samples and labels. This f^* can be considered a "regressor pattern" that always exists. By substituting Eq. (2) into Eq. (15), we get:

$$\begin{aligned}\mathbf{\Delta}_{ij} &= \lambda y_i + (1-\lambda)y_j - f^*(\lambda x_i + (1-\lambda)x_j) \\ &= \lambda f^*(x_i) + (1-\lambda)f^*(x_j) - f^*(\lambda x_i + (1-\lambda)x_j) \\ &= q(x_i, x_j, \lambda)\end{aligned} \tag{16}$$

Algorithm 1: Training process of DE-MIXUP

Input: The labeled dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and parameters $\{\alpha, \tau\}$
Ensure: An optimized regressor parameterized by $\mathbf{\Phi}, \mathbf{W}$.

```
Initialize $\{\mathbf{W}, \mathbf{B}\}$ randomly, and load a pre-trained deep feature encoder with $\mathbf{\Phi}$
Warm-up the regressor with $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$
For $t = 1$ to $N_{inner}$
    Draw a mini-batch $\Omega^{(t)}$ of $n_b$ labeled instances randomly
    Generate augmented instances of mixup
    Initialize $\mathbf{y}^{(t)} = \overline{\mathbf{y}}^{(t)}$, $\mathbf{\Delta^{(t)}} = \mathbf{0}$
    For $i = 1$ to $N_{inner}$
        Update $\mathbf{y}^{(t)}$ using Eq.(11)
        Update $\mathbf{\Delta^{(t)}}$ using Eq.(13)
        Update $\{\mathbf{\Phi}, \mathbf{W}, \mathbf{B}\}$ using stochastic gradients
        Update $\mathbf{\Theta}$ using Eq.(14)
    End For
End For
```

where $\mathbf{\Delta}_{ij}$ can be directly expressed using f^*. Since f^* always exists, it follows that $\mathbf{\Delta}_{ij}$ represents the "noise pattern" that must exist and is predictable.

Furthermore, given an augmented training dataset $\widetilde{D} = \{(\overline{x}_i, \overline{y}_i)\}_{i=1}^n$, we will easily learn the model parameter $\mathbf{W}$ using gradient descent on the square loss.

$$\hat{R}_S(W) = \frac{1}{n}\sum_{i=1}^n ||\mathbf{W}^T\overline{\Phi} - \overline{Y}||^2 \tag{17}$$

where $\overline{\Phi} = [h_{\mathbf{\Phi}}(\overline{x}_1), h_{\mathbf{\Phi}}(\overline{x}_2), \ldots, h_{\mathbf{\Phi}}(\overline{x}_n)] \in \mathbb{R}^{d\times n}$ and $\overline{Y} = [\overline{y}_1, \overline{y}_2, \ldots, \overline{y}_n] \in \mathbb{R}^n$. Then, we have the following important lemma with a detailed proof provided in Appendix A.

Lemma 1 *There exists* $\hat{\Phi} = (\overline{\Phi\Phi}^T)^{-1}\overline{\Phi}$ *that is the Moore-Penrose inverse of* $\overline{\Phi}^T$ *such that* $\mathbf{W}^* = \hat{\Phi}Y^*$ *wherein* $Y^* = [y_1^*, y_2^*, \ldots, y_n^*] \in \mathbb{R}^n$ *and* $\mathbf{B}^* = \hat{\Phi}\mathbf{\Delta}$ *wherein* $\mathbf{\Delta} = [\mathbf{\Delta}_1, \mathbf{\Delta}_2, \ldots, \mathbf{\Delta}_n] \in \mathbb{R}^n$. *It's natural to have the following closed form solution with learning rate* ϵ.

$$\mathbf{W}_t - \mathbf{W}^* = (\mathbf{W}_0 - \mathbf{W}^*)e^{-\frac{2\epsilon}{m}\overline{\Phi\Phi}^T t} + (I_d - e^{-\frac{2\epsilon}{m}\overline{\Phi\Phi}^T t})\mathbf{B}^*. \tag{18}$$

Remark 1 *Here,* $\mathbf{W}^*$ *and* $\mathbf{B}^*$ *represent the optimal parameters for the prediction functions* $f_{\mathbf{W}}$ *and* $g_{\mathbf{B}}$ *with square loss, respectively. Notably, the first term can be seen as the description of "regressor pattern" mentioned previously. The model parameters* $\mathbf{W}_t$ *converge to the regressor pattern* $\mathbf{W}^*$ *as* t *increases. The second term clearly demonstrates that the optimal parameters related to the noise pattern exist and can be learned through the deep features. This highlights the effectiveness of incorporating a noise estimation layer* $g_{\mathbf{B}}$*, parameterized by* $\mathbf{B}$ *with square loss to estimate the noise, as shown in Eq. (6).*

Error Bound

We will provide a boundary error analysis using DE-MIXUP. The square loss function $\mathcal{L}_2(\cdot,\cdot)$ is a common Lipschitz function, and assuming there exists a constant $\mathcal{C}_l$ such that $|\mathcal{L}_2(\cdot,\cdot)| \leq \mathcal{C}_l$. Additionally, the risk estimator of DE-MMIXUP can be expressed as follows:

$$R_{\mathbf{DE}}\left(f_{\mathbf{W}}, g_{\mathbf{B}}\right) = \mathbb{E}_{p(\overline{x},\overline{y})}\left[l_{DE}\left(f_{\mathbf{W}}\left(\overline{\Phi}\right), \overline{y}\right)\right] \tag{19}$$

Dataset		Size	Dimension	Type
In-distribution	Airfoil	1,503	5	Tabular
	NO2	500	7	Tabular
	Bike	17,379	16	Tabular
	Exchange-Rate	7,588	322	Time-series
	Electricity	26,304	6	Time-series
	Age	16,488	–	Image
	UTKface	24,106	–	Image
Out-of-distribution	RCF-MNIST	60,000	3×28×28	Image
	SkillCraft	3,395	20	Tabular
	Crime	306,094	16	Tabular

Table 1: Detailed information of datasets. – indicates that the sample feature dimensions are not uniform.

Furthermore, $\hat{R}_{\mathbf{DE}}\left(f_{\mathbf{W}}, g_{\mathbf{B}}\right)$ donate the empirical estimator of $R_{\mathbf{DE}}\left(f_{\mathbf{W}}, g_{\mathbf{B}}\right)$. Let $\hat{f}_{\mathbf{W}} = \arg\min_{f_{\mathbf{W}} \in \mathcal{F}} \hat{R}_{\mathbf{DE}}(f_{\mathbf{W}}, g_{\mathbf{B}})$ and $\hat{g}_{\mathbf{B}} = \arg\min_{g_{\mathbf{B}} \in \mathcal{G}} \hat{R}_{\mathbf{DE}}(f_{\mathbf{W}}, g_{\mathbf{B}})$. Additionally, $f_{\mathbf{W}}{}^* = \arg\min_{f_{\mathbf{W}} \in \mathcal{F}} R_{\mathbf{DE}}(f_{\mathbf{W}}, g_{\mathbf{B}})$ and $g_{\mathbf{B}}{}^* = \arg\min_{g_{\mathbf{B}} \in \mathcal{G}} R_{\mathbf{M}}(f_{\mathbf{W}}, g_{\mathbf{B}})$, where $\mathcal{F}$ and $\mathcal{G}$ are two independent hypothesis classes of the function. $\mathfrak{R}_n\left(\mathcal{F}\right)$ and $\mathfrak{R}_n\left(\mathcal{G}\right)$ denote the Rademacher complexity of augmented data with size n, respectively. Then we have the following theorem.

Theorem 1 *Given a augmented dataset $\widetilde{D} = \{(\overline{x_i}, \overline{y_i})\}_{i=1}^{n}$, and $\mathcal{L}_2$ is an $\rho - lipschitz$ function and is bounded by some constant $\mathcal{C}_l > 0$. For any $\delta > 0$, with the probability at least $1 - \delta$, the following holds for all $f_{\mathbf{W}} \in \mathcal{F}$ and $g_{\mathbf{B}} \in \mathcal{G}$.*

$$R_{\mathbf{DE}}\left(\hat{f}_{\mathbf{W}}, \hat{g}_{\mathbf{B}}\right) \leq R_{\mathbf{DE}}\left(f_{\mathbf{W}}{}^*, g_{\mathbf{B}}{}^*\right) + (4\alpha + 4)\rho\mathfrak{R}_n(\mathcal{F}) + 4\rho\mathfrak{R}_n(\mathcal{G}) + 2\mathcal{C}_l\sqrt{\frac{log\frac{2}{\delta}}{2n}} \tag{20}$$

The detailed proof is available in the Appendix B. The theorem 1 shows that the risk estimator $R_{\mathbf{DE}}$ on the augmented dataset $\widetilde{D}$ is bounded by the risk of regressor and noise estimation layer from $\mathcal{L}_2$. And as $n \to \infty$, $R_{\mathbf{DE}}\left(\hat{f}_{\mathbf{W}}, \hat{g}_{\mathbf{B}}\right) \to R_{\mathbf{DE}}\left(f_{\mathbf{W}}{}^*, g_{\mathbf{B}}{}^*\right)$, the overall convergence rate is characterized by $\mathcal{O}\left(\sqrt{\frac{1}{n}}\right)$.

Experiment

Experimental Settings

Datasets We evaluate the regression performance of DE-MIXUP **under the in-distribution cases** on 7 datasets, including tabular datasets, time-series datasets, and image datasets. The Airfoil dataset (Kooperberg 1997) includes aerodynamic and acoustic test results from airfoil blade sections in a wind tunnel. The NO2 emissions dataset (NO2) (Kooperberg 1997) is often used to predict th mount of air pollution at specific locations. The bike sharing dataset (Bike) (Fanaee-T and Gama 2014) records two years of bike rental counts in different environments. The Exchange-Rate dataset (Lai et al. 2018) contains daily exchange rates from eight countries, and the Electricity dataset (Lai et al. 2018) records electricity consumption from 321 customers. The AgeDB-DIR (Age) (Moschoglou et al. 2017) dataset and UTKface (Zhang, Song, and Qi 2017) are both used to train models for predicting an individual's age based on facial image features.

Additionally, we evaluate the generalization ability of DE-MIXUP **under the out-of-distribution cases** on 3 datasets. The RCFashion-MNIST (RCF-MNIST) (Yao et al. 2022) simulates distribution changes by reversing the spurious correlation between color and angle. The SkillCraft dataset (Blair et al. 2013) is used to train a model for predicting the LeagueIndex based on players' behavioral characteristics. The Crime dataset (Redmond 2009) contains a total of socio-economic and crime data from multiple communities. Its purpose is to predict violent crime rates per 1,000 population in unseen neighborhoods.

For clarity, we summarize the characteristics of datasets in Table 1.

Baselines We mainly compare DE-MIXUP with the commonly used mixup family methods and several modality-specific data augmentation methods. Specifically, the mixup family methods include mixup (Zhang et al. 2018), local mixup (Baena, Drumetz, and Gripon 2022), noisy-mixup (Lim et al. 2022), manifold mixup (mani mixup) (Verma et al. 2019), fair-mixup (Chuang and Mroueh 2021), mixupE (Zou et al. 2023), k-mixup (Greenewald et al. 2021), sk-mixup (Bouniot, Mozharovskyi, and d'Alché-Buc 2024), and c-mixup (Yao et al. 2022). Modality-specific methods include switchtab and vime for tabular data, time-warping and rotation for time-series data, and flipping, mask, and rotation for image data. Additionally for out-of-distribution cases, we employ 4 invariant learning methods, including IRM (Arjovsky et al. 2019), V-REx (Krueger et al. 2021), CORAL (Li et al. 2018b), and MLDG (Li et al. 2018a).

Implementation details We employ different deep feature encoders for different types of datasets, *i.e.,* a three-layer fully connected network for tabular datasets, the deep feature encoder of LST-Attn (Lai et al. 2018) for time-series datasets, and a pre-trained ResNet101[1] for image datasets. Additionally, we employ a single-layer fully connected network as the regression layer. During model training, we apply the Adam optimizer and the mean square error as the loss function. The batch size is fixed to 32.

For DE-MIXUP and DE-MMIXUP, we employ a single-layer fully connected network as the noise estimation layer. We warm-up DE-MIXUP and DE-MMIXUP with 50 epochs.

Evaluation metrics In the experiments, we measure the regression performance using Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE), where lower values indicate better performance. RMSE and MAPE are defined as $(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i^* - p_i)^2})$ and $(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i^* - p_i}{y_i^*}\right| \times 100)$, respectively, where y^* denotes the ground-truth response target.

Results under the In-distribution Cases

We present the empirical results for RMSE under the in-distribution cases in Table 2 and for MAPE in Table 4 (Ap-

[1] https://download.pytorch.org/models/

Method	Airfoil	NO2	Bike	Exchange-Rate	Electricity	Age	UTKface
benchmark	2.867 ± 0.239	0.522 ± 0.001	0.216 ± 0.039	0.0206 ± 0.004	0.057 ± 0.0004	13.778 ± 0.013	11.907 ± 0.189
Modality-specific data augmentation strategies							
+ switchtab	4.381 ± 0.459	0.586 ± 0.006	0.855 ± 0.063	–	–	–	–
+ vime	2.749 ± 0.002	0.522 ± 0.001	0.971 ± 0.094	–	–	–	–
+ time-warping	–	–	–	0.0205 ± 0.002	0.076 ± 0.0015	–	–
+ rotation	–	–	–	0.0889 ± 0.012	0.119 ± 0.0078	13.879 ± 0.038	11.892 ± 0.229
+ flipping	–	–	–	–	–	13.726 ± 0.006	11.633 ± 0.095
+ mask	–	–	–	–	–	13.857 ± 0.018	11.679 ± 0.103
Mixup family strategies							
+ mixup	3.761 ± 0.194	0.515 ± 0.006	0.496 ± 0.080	0.0208 ± 0.005	0.058 ± 0.0015	13.873 ± 0.006	11.827 ± 0.125
+ local mixup	4.165 ± 0.329	0.523 ± 0.005	0.511 ± 0.047	0.0330 ± 0.004	0.077 ± 0.0005	14.253 ± 0.075	12.638 ± 0.034
+ noisy-mixup	4.681 ± 0.085	0.557 ± 0.010	0.827 ± 0.269	0.0333 ± 0.006	0.081 ± 0.0007	14.587 ± 0.027	11.897 ± 0.101
+ mani mixup	3.056 ± 0.216	0.522 ± 0.015	0.289 ± 0.025	0.0235 ± 0.006	0.057 ± 0.0007	13.856 ± 0.012	11.872 ± 0.040
+ fair-mixup	2.917 ± 0.095	0.517 ± 0.011	0.241 ± 0.015	0.0629 ± 0.037	0.066 ± 0.0003	13.673 ± 0.044	11.771 ± 0.101
+ mixupE	3.775 ± 0.102	0.522 ± 0.010	0.453 ± 0.133	0.0200 ± 0.005	0.059 ± 0.0015	13.940 ± 0.073	11.959 ± 0.035
+ k-mixup	2.961 ± 0.226	0.518 ± 0.001	0.259 ± 0.011	0.0181 ± 0.002	0.093 ± 0.0130	14.626 ± 0.066	13.603 ± 0.089
+ sk-mixup	2.807 ± 0.162	0.544 ± 0.021	0.166 ± 0.009	0.0221 ± 0.001	0.069 ± 0.0006	13.774 ± 0.039	11.882 ± 0.163
+ c-mixup	2.795 ± 0.173	0.513 ± 0.012	0.214 ± 0.007	0.0199 ± 0.008	0.057 ± 0.0064	13.888 ± 0.025	12.016 ± 0.068
+ DE-MIXUP (Ours)	3.070 ± 0.184	0.508 ± 0.014	0.217 ± 0.006	0.0148 ± 0.005	0.059 ± 0.0057	13.739 ± 0.031	12.131 ± 0.072
+ DE-MMIXUP (Ours)	$\mathbf{2.332} \pm 0.127$	$\mathbf{0.498} \pm 0.008$	$\mathbf{0.141} \pm 0.007$	$\mathbf{0.0142} \pm 0.003$	$\mathbf{0.056} \pm 0.0043$	$\mathbf{13.662} \pm 0.019$	$\mathbf{11.564} \pm 0.061$

Table 2: Results of the average RMSE for three seeds under the in-distribution cases. "Benchmark" is the version without any data augmentation strategies. The best scores are indicated in bold.

pendix C). It can be significantly observed that our DE-MMIXUP performs significantly better than all methods, and DE-MIXUP also shows reliable performance. Specifically, DE-MMIXUP has a significant advantage over benchmark. For example, its RMSE decreases from 2.867 to 2.332, and its MAPE reduces from 1.723 to 1.450 on the Airfoil dataset. The results demonstrate that DE-MMIXUP achieves effective estimation of noise and thus ensures cleaner augmented data for the regression model. Meanwhile, DE-MMIXUP outperforms modality-specific data augmentation strategies, particularly on the Exchange-Rate and Electricity time-series datasets. For example, DE-MMIXUP reduces RMSE by 31.1% and MAPE by 39.3% on the Exchange-Rate dataset compared to time-warping. Additionally, modality-specific data augmentation strategies often perform worse than the benchmark, closely related to the noise inevitably introduced during the augmentation process.

Encouragingly, DE-MMIXUP also achieves better results compared to mixup and its variants. In particular, DE-MMIXUP outperforms the competitive c-mixup, reducing RMSE from 13.888 to 13.662 and MAPE from 32.147 to 31.638 on the Age dataset. This is because DE-MMIXUP offers a novel approach to eliminating noise from augmented labels by directly estimating them after applying mixup. This strategy reduces the impact of noisy labels on the model while preserving the diversity of augmented instances. Finally, it is worth noting that mixup and its simple variants are less effective and perform worse than the benchmark on all datasets. This underperformance is primarily due to mixup relying on the linear alignment assumption to generate augmented data, which introduces considerable noise (see more analysis in **Sections 3.2**). Furthermore, some mixup variants exhibit varying degrees of improvement. For example, mani-mixup enhances local linearity of the regression relationship by interpolating features at the hidden layer, thus effectively reducing noise in the augmented data.

Results under the Out-of-distribution Cases

We present the empirical results under the out-of-distribution cases in Table 3. DE-MMIXUP has the best performance, while DE-MIXUP also achieves competitive results. Compared to benchmark DE-MMIXUP has a significant improvement across all datasets. The RMSE and MAPE on the Crime dataset decreases from 0.143 to 0.130 and from 81.014 to 76.466. These improvements can be attributed to data augmentation, which provides a variety of samples for the regression model.

In comparison to invariant learning methods, DE-MMIXUP achieves greater advantages. For instance, it shows 9.04% and 4.75% improvements in RMSE and MAPE on the RCF-MNIST dataset compared to the V-REx method. Invariant learning methods extract key features from samples to achieve effective inference for unseen samples and are widely used in classification problems. However, the continuous labeling space in regression makes it more difficult to extract key features across samples, and relying solely on partial salient feature information often fails to ensure effective knowledge transfer for label prediction. In contrast, DE-MMIXUP effectively generates unseen out-of-distribution features by augmenting the dataset, which significantly improves the model's generalization ability.

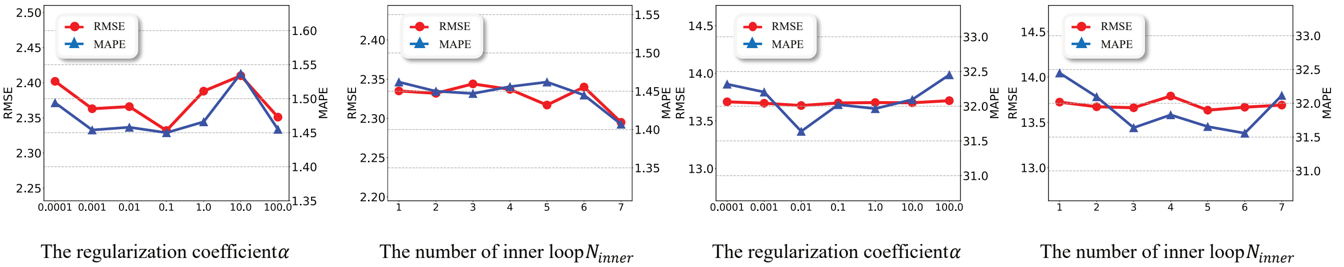
Finally, our method has the best performance compared to mixup and its variants, with RMSE decreasing from 5.813 to 5.571 on SkillCraft dataset compared to c-mixup. Since c-mixup uses the Gaussian kernel to select samples with similar labels for augmentation, it severely reduces the diversity of the augmented samples. This limitation hampers the robustness of the regression model when faced with out-of-distribution cases.

Parameter Analysis

We evaluated the effect of the regularization coefficient α and the number of inner loops on the model, as shown in

	RCF-MNIST		SkillCraft		Crime	
	RMSE	MAPE	RMSE	MAPE	RMSE	MAPE
benchmark	0.163 ± 0.012	742.010 ± 31.413	6.256 ± 0.124	10.729 ± 0.329	0.143 ± 0.0029	81.014 ± 4.251
Invariant learning methods						
+ IRM	0.161 ± 0.004	750.672 ± 34.162	8.662 ± 0.161	9.976 ± 0.484	0.132 ± 0.0041	84.623 ± 13.884
+ V-REx	0.166 ± 0.002	761.387 ± 35.096	9.785 ± 0.640	17.973 ± 1.425	0.131 ± 0.0048	88.904 ± 8.655
+ CORAL	0.158 ± 0.002	711.323 ± 30.390	6.073 ± 0.197	10.348 ± 0.271	0.130 ± 0.0043	87.668 ± 10.906
+ MLDG	0.163 ± 0.003	737.811 ± 27.856	8.023 ± 0.278	10.615 ± 0.461	0.133 ± 0.0052	83.594 ± 9.434
Mixup family strategies						
+ mixup	0.163 ± 0.002	821.317 ± 36.906	6.187 ± 0.253	10.887 ± 0.444	0.130 ± 0.0024	74.353 ± 6.743
+ local mixup	0.181 ± 0.015	779.445 ± 25.978	6.372 ± 0.546	11.551 ± 1.329	0.137 ± 0.0019	84.140 ± 7.614
+ noisy-mixup	0.161 ± 0.014	767.049 ± 27.816	8.473 ± 0.551	17.183 ± 1.337	0.138 ± 0.0004	88.170 ± 1.066
+ mani mixup	0.161 ± 0.013	**674.194** ± 24.423	6.010 ± 0.165	10.318 ± 0.406	0.131 ± 0.0031	86.414 ± 5.134
+ fair mixup	0.164 ± 0.010	728.415 ± 28.117	6.183 ± 0.145	10.610 ± 0.277	0.130 ± 0.0009	95.754 ± 8.996
+ mixupE	0.169 ± 0.017	776.832 ± 30.214	6.704 ± 0.162	11.808 ± 0.374	0.130 ± 0.0011	**73.966** ± 3.461
+ k-mixup	0.207 ± 0.023	730.001 ± 28.564	6.206 ± 0.591	14.363 ± 3.402	0.156 ± 0.0241	79.509 ± 3.072
+ sk-mixup	0.174 ± 0.011	702.684 ± 31.351	6.529 ± 0.154	11.362 ± 0.475	0.131 ± 0.0007	86.726 ± 4.127
+ c-mixup	0.159 ± 0.009	679.221 ± 27.451	5.813 ± 0.124	9.991 ± 0.270	0.132 ± 0.0081	78.871 ± 7.496
+ **DE-Mixup (Ours)**	0.157 ± 0.010	751.643 ± 34.823	5.776 ± 0.137	9.961 ± 0.311	0.136 ± 0.0073	75.291 ± 4.823
+ **DE-MMIXUP (Ours)**	**0.151** ± 0.007	725.236 ± 23.427	**5.571** ± 0.099	**9.780** ± 0.294	**0.130** ± 0.0076	76.466 ± 6.845

Table 3: Results of the average RMSE and MAPE for three seeds under the out-of-distribution cases. "Benchmark" is the version without any data augmentation strategies. The best scores are indicated in bold.



(a) Results on the Airfoil dataset.

(b) Results on the Age dataset.

Figure 1: DE-MMIXUP performance when varying the regularization factor α and the number of inner loops N_{inner}.

Figure 1. Initially, the DE-MMIXUP's performance improves as α increases, demonstrating its effectiveness in correcting noise. However, when α becomes too large, it disrupts the model's noise estimation, leading to a decline in performance. The best results were obtained with α values of 0.1 and 0.01 for the Airfoil and Age datasets, respectively.

Additionally, as the number of inner loops increased, the model's performance improved on both datasets. This indicates that DE-MMIXUP can more effectively learn the noise between true labels and augmented labels over multiple loops, thereby efficiently correcting noise and preventing the regression model from overfitting to noisy instances. It is worth noting that after a certain number of iterations, the performance improvement of DE-MMIXUP on both datasets is not significant. Therefore, to balance model efficiency and performance, we selected 2 and 3 inner loops for DE-MMIXUP on the airfoil and age datasets, respectively.

Ablation Study

We will evaluate the impact of the noise estimation layer and the manifold mixup operation on the performance of DE-MMIXUP. The specific results are presented in Tables 2 to 4, comparing mixup with DE-MIXUP and DE-MIXUP with DE-MMIXUP, respectively. The results clearly show that the noise estimation layer effectively reduces augmented label noise. Additionally, the manifold mixup operation further regularizes the augmented samples, and both components contribute to improving the performance of regressor.

Conclusion

In this paper, we focus on label noise caused by mixup in regression tasks and propose a novel method called DE-MIXUP (DE-MMIXUP), which leverages an auxiliary noise estimation task to correct it. We conduct extensive experiments under both the in-distribution and out-of-distribution cases, showing that DE-MIXUP and DE-MMIXUP significantly outperform mixup and its variants, modality-specific data augmentation strategies, and invariant learning methods. The results demonstrate that our methods can not only accurately estimate noise but also provide clean instances for training the regressor, thereby enhancing the model's stability and robustness across different scenarios.

Acknowledgments

We acknowledge support for this project from the National Science and Technology Major Project (No.2021ZD0112500) and the National Natural Science Foundation of China (No.62276113).

References

Arjovsky, M.; Bottou, L.; Gulrajani, I.; and Lopez-Paz, D. 2019. Invariant Risk Minimization. *arXiv preprint arXiv:1907.02893*.

Baek, K.; Bang, D.; and Shim, H. 2021. GridMix: Strong regularization through local context mapping. *Pattern Recognition*, 109: 107594.

Baena, R.; Drumetz, L.; and Gripon, V. 2022. Preventing Manifold Intrusion With Locality: Local Mixup. *arXiv preprint arXiv:2201.04368*.

Blair, M.; Thompson, J.; Henrey, A.; and Chen, B. 2013. Skillcraft1 Master Table Dataset. *UCI Machine Learning Repository*.

Bouniot, Q.; Mozharovskyi, P.; and d'Alché-Buc, F. 2024. Tailoring Mixup to Data for Calibration. *arXiv preprint arXiv:2311.01434*.

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1): 1–122.

Cao, C.; Zhou, F.; Dai, Y.; Wang, J.; and Zhang, K. 2024. A Survey of Mix-based Data Augmentation: Taxonomy, Methods, Applications, and Explainability. *ACM Computing Surveys*, 57: 1–38.

Castells, T.; Weinzaepfel, P.; and Revaud, J. 2020. SuperLoss: a generic loss for robust curriculum learning. In *International Conference on Neural Information Processing Systems*, 4308–4319.

Chou, H.; Chang, S.; Pan, J.; Wei, W.; and Juan, D. 2021. Remix: Rebalanced Mixup. In *European Conference on Computer Vision*, 95–110.

Chuang, C.; and Mroueh, Y. 2021. Fair Mixup: Fairness via Interpolation. In *International Conference on Learning Representations*.

de Vente, C.; Vos, P.; Hosseinzadeh, M.; Pluim, J.; and Veta, M. 2020. Deep Learning Regression for Prostate Cancer Detection and Grading in Bi-Parametric MRI. *IEEE Transactions on Biomedical Engineering*, 68: 374–383.

Fanaee-T, H.; and Gama, J. 2014. Event Labeling Combining Ensemble Detectors and Background Knowledge. *Progress in Artificial Intelligence*, 2: 113–127.

Feng, S. Y.; Gangal, V.; Wei, J.; Chandar, S.; Vosoughi, S.; Mitamura, T.; and Hovy, E. 2024. A Survey of Data Augmentation Approaches for NLP. *arXiv preprint arXiv:2105.03075v5*.

Franchi, G.; Belkhir, N.; Ha, M. L.; Hu, Y.; Bursuc, A.; Blanz, V.; and Yao, A. 2021. Robust Semantic Segmentation with Superpixel-Mix. *arXiv preprint arXiv:2108.00968*.

Gillberg, J.; Marttinen, P.; Pirinen, M.; Kangas, A. J.; Soininen, P.; Ali, M.; Havulinna, A. S.; Järvelin, M.; Ala-Korpela, M.; and Kaski, S. 2016. Multiple Output Regression with Latent Noise. *Journal of Machine Learning Research*, 17: 1–35.

Greenewald, K.; Gu, A.; Yurochkin, M.; Solomon, J.; and Chien, E. 2021. k-Mixup Regularization for Deep Learning via Optimal Transport. *arXiv preprint arXiv:2106.02933*.

Guo, H. 2020. Nonlinear mixup: Out-of-manifold data augmentation for text classification. In *the AAAI Conference on Artificial Intelligence*, 4044–4051.

Guo, H.; Mao, Y.; and Zhang, R. 2019. MixUp as Locally Linear Out-of-Manifold Regularization. In *the AAAI Conference on Artificial Intelligence*, 3714–3722.

Han, X.; Jiang, Z.; Liu, N.; and Hu, X. 2022. G-Mixup: Graph Data Augmentation for Graph Classification. In *International Conference on Machine Learning*, 8230–8248.

Hong, M.; Choi, J.; and Kim, G. 2021. StyleMix: Separating Content and Style for Enhanced Data Augmentation. In *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14862–14870.

Hwang, S.; Kim, M.; and Whang, S. E. 2024. RC-Mixup: A Data Augmentation Strategy against Noisy Data for Regression Tasks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1155–1165.

Hwang, S.; and Whang, S. E. 2021. RegMix: Data Mixing Augmentation for Regression. *arXiv preprint arXiv:2106.03374*.

Kim, C. D.; Moon, S.; Moon, J.; Woo, D.; and Kim, G. 2024. Sample Selection via Contrastive Fragmentation for Noisy Label Regression. In *Conference on Neural Information Processing Systems*.

Kim, J.; Choo, W.; and Song, H. O. 2020. Puzzle Mix: Exploiting Saliency and Local Statistics for Optimal Mixup. In *International Conference on Machine Learning*, 5275–5285.

Kooperberg, C. 1997. Statlib: an Archive for Statistical Software, Datasets, and Information. *The American Statistician*, 51(1): 98.

Kou, Z.; Wang, J.; Jia, Y.; Liu, B.; and Geng, X. 2025a. Instance-Dependent Inaccurate Label Distribution Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1): 1425–1437.

Kou, Z.; Xuan, H.; Zhu, J.; Wang, H.; Xie, M.-k.; Wang, C.; Wang, J.; Jia, Y.; and Geng, X. 2025b. Tail-Aware Reconstruction of Incomplete Label Distributions with Low-Rank and Sparse Modeling. *IEEE Transactions on Circuits and Systems for Video Technology*, 1–1.

Krueger, D.; Jacobsen, J.; Zhang, A.; Binas, J.; Zhang, D.; Priol, R. L.; and Courville, A. 2021. Out-of-Distribution Generalization via Risk Extrapolation. In *International Conference on Machine Learning*, 5815–5826.

Lai, G.; Chang, W.; Yang, Y.; and Liu, H. 2018. Modeling Long-and short-term Temporal Patterns with Deep Neural Networks. In *International ACM SIGIR Conference on Research & Development in Information Retrieval*, 95–104.

Li, D.; Yang, Y.; Song, Y.; and Hospedales, T. M. 2018a. Learning to Generalize: Meta-Learning for Domain Generalization. In *The AAAI Conference on Artificial Intelligence*.

Li, H.; Pan, S. J.; Wang, S.; and Kot, A. C. 2018b. Domain Generalization with Adversarial Feature Learning. In *the IEEE Conference on Computer Vision and Pattern Recognition*, 5400–5409.

Lim, S. H.; Erichson, N. B.; Utrera, F.; Xu, W.; and Mahoney, M. W. 2022. Noisy Feature Mixup. In *International Conference on Learning Representations*.

Liu, Z.; Wang, Z.; Guo, H.; and Mao, Y. 2023. Over-training with Mixup may Hurt Generalization. In *arXiv preprint arXiv:2303.01475*.

Mai, Z.; Hu, G.; Chen, D.; Shen, F.; and Shen, H. T. 2022. MetaMixUp: Learning Adaptive Interpolation Policy of MixUp With Metalearning. *IEEE Transactions on Neural Networks and Learning Systems*, 33: 3050–3064.

Moschoglou, S.; Papaioannou, A.; Sagonas, C.; Deng, J.; Kotsia, I.; and Zafeiriou, S. 2017. Agedb: The First Manually Collected, In-the-wild Age Database. In *the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 51–59.

Redmond, M. 2009. Communities and Crime. *UCI Machine Learning Repository*.

Schneider, N.; Goshtasbpour, S.; and Perez-Cruz, F. 2023. Anchor data augmentation. In *International Conference on Neural Information Processing Systems*, 74890–74902.

Sial, H. A.; Baldrich, R.; Vanrell, M.; and Samaras, D. 2020. Light Direction and Color Estimation from Single Image with Deep Regression. *arXiv preprint arXiv:2009.08941*.

Uddin, A. F. M. S.; Monira, M. S.; Shin, W.; Chung, T.; and Bae, S. 2021. SaliencyMix: A Saliency Guided Data Augmentation Strategy for Better Regularization. In *International Conference on Learning Representations*.

Venkataramanan, S.; Kijak, E.; Amsaleg, L.; and Avrithis, Y. 2020. AlignMixup: Improving Representations by Interpolating Aligned Features. In *the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5275–5285.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold Mixup: Better Representations by Interpolating Hidden States. In *International Conference on Machine Learning*, 6438–6447.

Xu, M.; Yoon, S.; Fuentes, A.; and Park, D. S. 2023. A Comprehensive Survey of Image Augmentation Techniques for Deep Learning. *Pattern Recognition*, 137: 109347.

Yao, H.; Wang, Y.; Zhang, L.; Zou, J.; and Finn, C. 2022. C-Mixup: Improving Generalization in Regression. In *Conference on Neural Information Processing Systems*, 3361–33767.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *the IEEE/CVF International Conference on Computer Vision*, 6023–6032.

Zha, D.; Bhat, Z. P.; Lai, K.-H.; Yang, F.; Jiang, Z.; Zhong, S.; and Hu, X. 2024. Data-centric Artificial Intelligence: A Survey. *ACM Computing Surveys*, 57: 1–42.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *International Conference on Learning Representations*.

Zhang, L.; Aggarwal, C.; and Qi, G. 2017. Stock Price Prediction via Discovering Multi-Frequency Trading Patterns. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2141–2149.

Zhang, Z.; Song, Y.; and Qi, H. 2017. Age Progression/Regression by Conditional Adversarial Autoencoder. In *the IEEE Conference on Computer Vision and Pattern Recognition*, 5810–5818.

Zou, Y.; Verma, V.; Mittal, S.; HohTang, W.; HieuPham; Kannala, J.; Bengio, Y.; Solin, A.; and Kawaguchi, K. 2023. Mixupe: Understanding and Improving Mixup from Directional Derivative Perspective. In *Conference on Uncertainty in Artificial Intelligence*, 2597–2607.