RankUp: Boosting Semi-Supervised Regression with an Auxiliary Ranking Classifier

Pin-Yen Huang
Academia Sinica
Taipei, Taiwan
pyhuang97@gmail.com

Szu-Wei Fu
NVIDIA
Taipei, Taiwan
szuweif@nvidia.com

Yu Tsao
Academia Sinica
Taipei, Taiwan
yu.tsao@citi.sinica.edu.tw

Abstract

State-of-the-art (SOTA) semi-supervised learning techniques, such as FixMatch and it's variants, have demonstrated impressive performance in classification tasks. However, these methods are not directly applicable to regression tasks. In this paper, we present RankUp, a simple yet effective approach that adapts existing semi-supervised classification techniques to enhance the performance of regression tasks. RankUp achieves this by converting the original regression task into a ranking problem and training it concurrently with the original regression objective. This auxiliary ranking classifier outputs a classification result, thus enabling integration with existing semi-supervised classification methods. Moreover, we introduce regression distribution alignment (RDA), a complementary technique that further enhances RankUp's performance by refining pseudo-labels through distribution alignment. Despite its simplicity, RankUp, with or without RDA, achieves SOTA results in across a range of regression benchmarks, including computer vision, audio, and natural language processing tasks. Our code and log data are open-sourced at `https://github.com/pm25/semi-supervised-regression`.

1 Introduction

The effectiveness of deep learning models heavily depends on the availability of labeled data. However, obtaining labeled data can be challenging in various scenarios. For instance, tasks like quality assessment often require multiple human annotators to label a single data point [15, 8, 19], resulting in a labor-intensive and time-consuming process. In domains where expert annotation is frequently required, such as medical data, the cost of acquiring labeled data can be extremely expensive [14, 21, 36]. To address these challenges, semi-supervised learning provides a powerful approach to reduce reliance on labeled data for training deep learning models [22, 34, 17, 24, 7]. By effectively leveraging the unlabeled data during model training, semi-supervised learning provides a means to enhance model performance while minimizing the need for extensive labeled data.

Recent state-of-the-art (SOTA) semi-supervised learning methods, such as FixMatch and its variants, use a confidence threshold technique to obtain high-quality pseudo-labels [22, 34, 27, 4, 31]. This approach involves generating pseudo-labels from unlabeled data and then filtering out those with low confidence scores. The model is then trained to produce predictions consistent with these high-quality pseudo-labels. Despite its success across various classification tasks, directly applying this technique to regression tasks encounters several challenges. First, unlike classification models, regression models typically lack confidence measures for their predictions, making the confidence threshold technique unfeasible. Additionally, one of the motivations behind using pseudo-labels is to increase the model's confidence in its predictions for unlabeled data, based on the low-density assumption [6, 25]. However, as there are no confidence measures in the predictions of regression models, relying on the low-density assumption and increasing the confidence of unlabeled data becomes unfeasible.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-3413

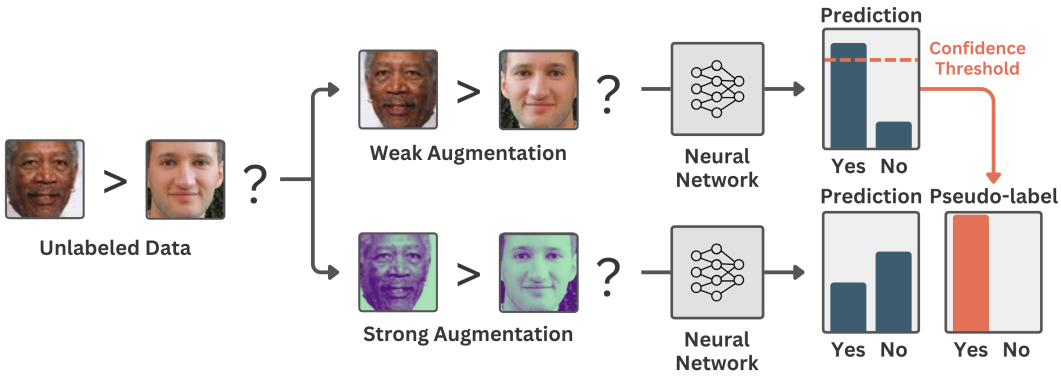

Figure 1: Illustration of using FixMatch on the Auxiliary Ranking Classifier (ARC). This diagram uses the age estimation task as an example, where the goal is to predict the age of a person in an image. The auxiliary ranking classifier transforms this task into a ranking problem by comparing two images to determine which person is older. (Image sourced from the UTKFace dataset [37]).

In this paper, we introduce *RankUp*, a simple yet effective semi-supervised regression framework that leverages existing semi-supervised classification methods. RankUp achieves this by using an auxiliary ranking classifier, which concurrently solves a ranking task alongside the original regression task. The ranking task is derived from the original regression problem, where the objective is to compare the labels of pairs of samples to determine their relative rank (i.e., which one is larger or smaller). Since ranking problem is a type of classification problem, existing semi-supervised classification methods can be applied to assist in training the auxiliary ranking classifier (see Fig. 1).

Our empirical results demonstrate that enhancing the performance of the auxiliary ranking classifier also improves the performance of the original regression task, as measured by metrics, such as mean absolute error (MAE) [29], coefficient of determination (R^2) [23], and Spearman rank correlation coefficient (SRCC) [33]. Furthermore, we show that applying existing semi-supervised classification methods to the auxiliary ranking classifier can effectively utilize unlabeled data, leading to further improvements in the classifier's performance. This improvement, in turn, translates to enhanced performance in the original regression task, showcasing the potential of applying semi-supervised classification techniques to enhance regression models.

One of the key advantages of using the auxiliary ranking classifier is its ability to enhance the ranking relationship of pseudo-labels. Building upon this effect, we propose a novel *Regression Distribution Alignment* (RDA) method, designed to further improve RankUp's performance by refining the distribution of regression pseudo-labels. RDA adjusts the distribution of these pseudo-labels to better align with the true underlying distribution of the unlabeled data. This approach assumes that the distributions of labeled and unlabeled data are similar, allowing us to estimate the distribution of the unlabeled data based on that of the labeled data distribution. This assumption holds true in many cases, especially when labeled data are randomly sampled from the same pool as the unlabeled data. By aligning these distributions, RDA improves the quality of the pseudo-labels, ultimately leading to better model performance when training with these refined pseudo-labels.

Our experimental results demonstrate that RankUp, even without RDA, achieves state-of-the-art (SOTA) results across a variety of regression datasets, including tasks in computer vision, audio, and natural language processing. Moreover, integrating RDA with RankUp provides an additional performance boost, leading to the highest performance observed in our experiments. For example, RankUp alone achieves at least a 13% improvement in MAE and a 28% improvement in R^2 compared to SOTA methods on the image age estimation dataset (UTKFace) with 50 labeled samples. The addition of RDA further boosts these results by an additional 6% and 7% in MAE and R^2, respectively. The empirical results of our experiments demonstrate that existing semi-supervised classification methods can be effectively leveraged to improve the performance of semi-supervised regression tasks. **These findings bridge the gap between future research in semi-supervised regression and classification, paving the way for further advancements in the field.**

2 Related Works

In this section, we review related research in semi-supervised learning. We categorize the literature into two groups: methods applicable to regression tasks, which will be discussed in Section 2.1, and methods applicable only to classification tasks, detailed in Section 2.2.

2.1 Semi-Supervised Regression

In semi-supervised regression, methods commonly rely on the smoothness assumption [6, 25], which suggests that nearby data points in the feature space should share similar labels. Consistency regularization is a popular technique employed to achieve this assumption. It encourages models to generate consistent predictions for slightly perturbed data.

For example, the Π-model [17] applies data augmentation to unlabeled data and minimizes the squared difference between the predictions of the augmented data and their original counterparts. Techniques like Mean Teacher [24] involve model-weight ensembling to align the predictions of the model with its ensemble counterpart. Similarly, UCVME [10] employs a bayesian neural network to ensure consistency in uncertainty predictions across co-trained models. Additionally, CLSS [11] utilizes contrastive learning to encourage features of similar labels to be closer together.

2.2 Semi-Supervised Classification

In semi-supervised classification, in addition to the smoothness assumption, another commonly relied-upon assumption is the low-density assumption [6, 25]. This assumption suggests that a classifier's decision boundary should ideally pass through low-density regions in the feature space. Pseudo-labeling [18] is a common approach used to achieve this assumption, where the highest probability class predictions on unlabeled data are utilized as pseudo-labels for training. By incorporating pseudo-labels, the model's confidence in predicting unlabeled data is increased, effectively pushing the decision boundary away from high-density regions towards low-density regions.

Recent SOTA semi-supervised learning methods combine pseudo-labeling with consistency regularization to achieve both the low-density and smoothness assumptions, leading to improved performance. For example, MixMatch [3] utilizes a mixup [35] technique and averages predictions from multiple augmented instances to ensure consistency, while also using a sharpening technique to boost prediction confidence on unlabeled data. Similarly, FixMatch [22] builds on this concept by generating high-quality pseudo-labels from weakly augmented unlabeled data using a confidence threshold and enforcing consistency between weakly and strongly augmented versions of the same input.

Despite the success of these methods on classification tasks. The low-density assumption doesn't directly translate to regression tasks, as regression models lack explicit confidence measures and decision boundaries like those in classification models. As a result, existing semi-supervised learning methods based on the low-density assumption cannot be directly applied in regression settings.

3 Method

The proposed framework, RankUp, introduces two additional components: ARC and RDA. The design of ARC is inspired by RankNet [5]. To provide a clear understanding of the ARC's implementation, we first present background information on RankNet in Section 3.1. Subsequently, we will detail the implementation of ARC in Section 3.2 and introduce RDA in Section 3.3. Furthermore, we propose a warm-up scheme and techniques for reducing the computational time of RDA in Sections 3.4 and 3.5, respectively. Lastly, we outline the complete RankUp framework in Section 3.6.

3.1 Background: RankNet

RankNet [5] is a deep learning model designed to predict the relevance scores of documents. The core idea behind RankNet is the use of a pairwise ranking loss. It compares two samples and predicts their relative ranking (i.e., which document is more relevant). This approach effectively transforms the relevance score prediction task into a pairwise classification problem. In the following, we will provide a detailed explanation of how the pairwise ranking prediction is performed and how the corresponding loss is calculated.

 https://doi.org/10.52202/079017-3413

Pairwise Ranking Prediction. The output of RankNet is a single scalar value indicating the ranking score of the sample, where a higher score indicates greater relevance. To obtain the pairwise ranking prediction, two samples are fed separately into the model to get their respective ranking scores. The difference between these scores is then passed through a sigmoid function, which generates a prediction in the range [0, 1]. This prediction indicates the likelihood that the first sample is more relevant than the second. If the output is greater than 0.5, the model predicts that the first sample has higher relevance; if the output is less than 0.5, the second sample is considered more relevant. Mathematically, for two samples, x_i and x_j, and the RankNet model g, the formula to obtain the pairwise ranking prediction p_{ij} is as follows:

$$p_{ij} = \text{sigmoid}(g(x_i) - g(x_j)) = \frac{1}{1 + e^{-(g(x_i) - g(x_j))}} \tag{1}$$

Here, $g(x_i)$ and $g(x_j)$ represent the ranking scores for samples x_i and x_j, respectively. A higher value of p_{ij} indicates a higher likelihood that the ranking of x_i will be higher than that of x_j.

Pairwise Ranking Loss. The pairwise ranking loss is calculated by comparing the model's predicted pairwise ranking p_{ij} with the ground truth label y_{ij}. The label y_{ij} indicates the true relative ranking between samples x_i and x_j. Specifically, $y_{ij} = 1$ indicates that sample x_i is ranked higher than sample x_j, $y_{ij} = 0$ indicates that sample x_i is ranked lower than x_j, and $y_{ij} = 0.5$ suggests the two samples are equally ranked. Since this is fundamentally a binary classification task, the pairwise ranking loss is calculated using the cross-entropy loss function. Mathematically, the pairwise ranking loss for a batch of data is defined as follows:

$$\ell_{ranknet} = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \text{CE}(y_{ij},\ p_{ij}) \tag{2}$$

Here, N denotes the batch size, CE is the cross-entropy loss function, p_{ij} is the predicted pairwise ranking between samples x_i and x_j, and y_{ij} is the corresponding ground truth label. The loss iterates through all possible pairs of samples in the batch to calculate the average loss for the entire batch.

3.2 Auxiliary Ranking Classifier (ARC)

The *Auxiliary Ranking Classifier* (ARC) is designed to solve a ranking task alongside the primary regression task. It can be easily integrated into existing regression model architectures like ResNet [13], BERT [12], or Whisper [20]. ARC is implemented as an additional output layer that shares the same hidden layers with the original regression model. This transforms the model into a multi-task architecture with two output tasks: the original output header continues to provide the regression output, while ARC generates a ranking score for the sample.

The core idea behind ARC is to transform the original regression task into a multi-class classification problem, allowing existing semi-supervised classification methods to assist in its training. To achieve this, ARC's design is inspired by RankNet, which can effectively convert the regression task into a binary classification task. However, since a multi-class classification output is required, we introduce two key modifications to RankNet to adapt it for this purpose:

1. The scalar output value of RankNet is changed to a two-class output, where each output class indicates which sample in a pair has a relatively greater ranking score.
2. The sigmoid function is replaced with softmax, which converts the model's output into a multi-class classification probability distribution.

Specifically, for the auxiliary ranking classifier r, which outputs a two-class output, the formula to obtain the pairwise ranking prediction $\hat{p}_{ij}$ of two data samples, x_i and x_j, is as follows:

$$\hat{p}_{ij} = \text{softmax}(r(x_i) - r(x_j)) \tag{3}$$

Here, $\hat{p}_{ij}$ represents a two-class prediction that indicates which sample in the pair has a relatively higher regression label. The loss calculation for ARC remains the same as described in Equation 2.

This output format enables the integration of existing semi-supervised classification methods. We utilize FixMatch [22] as the semi-supervised classification technique for training ARC. An illustrative example of applying FixMatch to ARC can be found in Figure 1, with further details in Algorithm 1.

Algorithm 1 Auxiliary Ranking Classifier (with FixMatch)

Input: Labeled batch $X = \{ (x_i,\ y_i) \}_{i=1}^{N_{lb}}$, unlabeled batch $U = \{ u_i \}_{i=1}^{N_{ulb}}$, confidence threshold τ, unlabeled loss weight ω_{ulb}, weak augmentation A_w, strong augmentation A_s

$\ell_{lb} = \frac{1}{(N_{lb})^2} \sum_{i=1}^{N_{lb}} \sum_{j=1}^{N_{lb}} \text{CE}\Big(\text{softmax}\big(r(A_w(x_i)) - r(A_w(x_j))\big),\ \mathbb{1}\{y_i > y_j\}\Big)$ { Compute cross-entropy labeled loss }
$\ell_{ulb} = 0$ { Initialize unlabeled loss }
for $i = 1$ **to** N_{ulb} **do**
  for $j = 1$ **to** N_{ulb} **do**
    $\hat{p}_{ij}^{w} = \text{softmax}(r(A_w(u_i)) - r(A_w(u_j)))$ { Predict weak pairwise ranking }
    $\hat{p}_{ij}^{s} = \text{softmax}(r(A_s(u_i)) - r(A_s(u_j)))$ { Predict strong pairwise ranking }
    $\ell_{ulb} = \ell_{ulb} + \mathbb{1}\{\max(\hat{p}_{ij}^{w}) > \tau\}\ \text{CE}(\arg\max(\hat{p}_{ij}^{w}),\ \hat{p}_{ij}^{s})$ { Accumulate unlabeled loss }
  end for
end for
$\ell_{ulb} = \frac{1}{(N_{ulb})^2} \ell_{ulb}$ { Average unlabeled loss }
return $\ell_{\text{arc}} = \ell_{lb} + \omega_{ulb} \cdot \ell_{ulb}$

3.3 Regression Distribution Alignment (RDA)

Distribution alignment is a commonly used technique in semi-supervised classification [2, 16, 28, 4], where pseudo-labels are refined by aligning their distribution with that of the labeled data. Training semi-supervised models with these refined pseudo-labels can lead to performance improvements. However, existing distribution alignment methods are designed for classification tasks involving discrete label distributions, making them unsuitable for regression settings. Moreover, applying distribution alignment to ARC is impractical, as the two output classes always have equal proportions. To address these challenges, we propose *Regression Distribution Alignment* (RDA), enabling the direct application of distribution alignment to regression tasks.

The RDA process involves three key steps: (1) extracting the labeled data distribution, (2) generating the pseudo-label distribution, and (3) aligning the pseudo-label distribution with the labeled data distribution. These steps correspond to the orange, blue, and yellow parts of Figure 2, respectively.

Step 1: Extracting the Labeled Data Distribution. The labeled data is sorted according to its label values. To ensure a one-to-one correspondence with the pseudo-labels, the labeled data distribution must contain the same number of data points as the pseudo-label set. We use linear interpolation to extend the labeled distribution to match the size of the pseudo-labels.

Step 2: Generating the Pseudo-Label Distribution. The model generates pseudo-labels for all the unlabeled data. These pseudo-labels are then sorted by their values, either in ascending or descending order, as long as the sorting direction is consistent with that of the labeled data distribution.

Step 3: Aligning the Distributions. Once both distributions have been sorted and resized to the same size, the alignment is performed by replacing each pseudo-label value with its corresponding value from the labeled data distribution.

In each training iteration, RDA is applied to refine the pseudo-labels. The loss is then computed between the model's predictions and the RDA-aligned pseudo-labels to minimize their difference. For an unlabeled data point u_i with its corresponding regression prediction $\hat{y}_i$ and RDA-aligned pseudo-label $\tilde{y}_i$, the RDA loss ℓ_{rda} for a batch of unlabeled data is defined as:

$$\ell_{\text{rda}} = \frac{1}{N_{ulb}} \sum_{i=1}^{N_{ulb}} L_{reg}\left(\hat{y}_i,\ \tilde{y}_i\right) \tag{4}$$

Here, N_{ulb} denotes the batch size of unlabeled data, L_{reg} represents a regression loss function (e.g., MAE, MSE).

 https://doi.org/10.52202/079017-3413

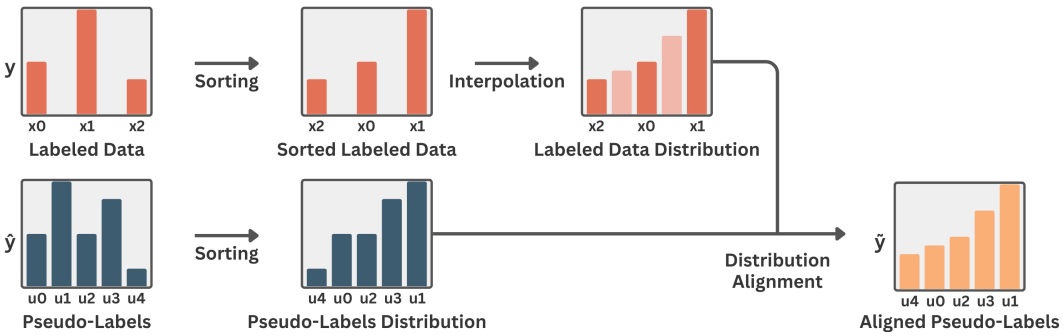

Figure 2: Illustration of RDA: This example includes three labeled data pairs $\{(x_i,\ y_i)\}_{i=0}^{2}$ and five unlabeled data points with corresponding pseudo-labels $\{(u_i,\ \hat{y}_i)\}_{i=0}^{4}$. Each data pair is represented by a single bar in the graph. The x-axis indicates the sample indices, while the y-axis represents their corresponding regression label values. The orange bars demonstrate the process of obtaining the labeled data distribution, the blue bars illustrate how the pseudo-label distribution is formed, and the yellow bars show the aligned pseudo-labels after applying RDA.

The design of RDA is based on two key assumptions. First, it assumes that the distributions of labeled and unlabeled data are similar, which is often true since labeled data is typically randomly sampled from the unlabeled pool. Second, it assumes that the ranking of the pseudo-labels is reasonably accurate. Integrating RDA with ARC can reinforce this assumption, as ARC enhances the ranking relationships of pseudo-labels. Both assumptions are crucial for ensuring that RDA works properly.

3.4 Warm-Up Scheme for RDA

In the early stages of training, the pseudo-label rankings may be poorly predicted, which can degrade the quality of the pseudo-labels refined through RDA. To address this, we introduce a linear warm-up scheme to stabilize the RDA process. The adjusted RDA loss, ℓ'_{rda}, is defined as follows:

$$\ell'_{\text{rda}} = \min\left(\frac{iter}{\alpha_{\text{warm}}},\ 1.0\right) \cdot \ell_{\text{rda}} \tag{5}$$

Here, $iter$ denotes the current training iteration, and α_{warm} is a hyperparameter that controls the duration of the warm-up phase. The $\min$ function ensures that the warm-up factor does not exceed 1.0, smoothly transitioning the model toward the full effect of ℓ_{rda}.

3.5 Reducing Computational Time of RDA

Applying RDA can be computationally expensive, as it requires inference all unlabeled data and sorting all pseudo-labels at every training iteration. This significantly increases the computational load compared to the original training process, especially when dealing with a large volume of unlabeled data, making the implementation of RDA impractical. To mitigate this challenge, we propose several techniques aimed at reducing the computational burden of RDA.

Pseudo-label table. RDA creates a table of the same size as the unlabeled dataset. This table stores the model's predicted pseudo-labels for each instance of unlabeled data. For each training iteration, the model generates new pseudo-labels, which are stored and updated within this table. This approach eliminates the need to rerun inference on all unlabeled data when applying RDA, as it only requires a simple lookup from the pseudo-label table.

Applying RDA only every T steps. To further reduce computational costs, RDA is applied only every T steps, where T is a hyperparameter. This is achieved by creating a second table of the same size as the unlabeled dataset, which stores the previously aligned results of the pseudo-labels generated by applying RDA. Between RDA updates, the model uses these stored aligned pseudo-labels, thereby avoiding the need to run RDA in every iteration. This strategy effectively reduces the computational cost associated with the RDA process to $1/T$.

3.6 Putting It All Together - RankUp

We introduce the term *RankUp* to describe our proposed framework, which integrates two key components: ARC and RDA. The use of RDA is optional, depending on whether its underlying assumptions are satisfied. The final loss for RankUp is a combination of the regression loss and the ARC loss. The regression loss consists of the original labeled regression loss plus the unlabeled RDA loss. Specifically, the RankUp loss ℓ_{rankup} is defined as follows:

$$\ell_{\text{rankup}} = (\ell_{\text{reg}} + \omega_{\text{rda}} \cdot \ell'_{\text{rda}}) + (\omega_{\text{arc}} \cdot \ell_{\text{arc}}) \tag{6}$$

In this equation, ℓ_{reg} represents the loss from the original labeled regression task, while ℓ'_{rda} denotes the RDA loss, as defined in Equation 5. The hyperparameter ω_{rda} controls the weight of the RDA loss. If RDA is not employed, the term $\omega_{\text{rda}} \cdot \ell'_{\text{rda}}$ can be excluded from the equation. The term ℓ_{arc} corresponds to the loss from the ARC module, as detailed in Algorithm 1. Additionally, the hyperparameter ω_{arc} regulates the weight of the ARC loss.

4 Experiments

In this section, we evaluate RankUp's performance across various tasks. The experimental settings are described in Section 4.1. The main results for RankUp under different label configurations are presented in Section 4.2, while Section 4.3 provides additional results on audio and text datasets. Section 4.4 explores the use of alternative semi-supervised classification methods in place of FixMatch. Lastly, we discuss potential reasons why the smoothness and low-density assumptions are also effective for regression tasks in Section 4.5.

4.1 Settings

Evaluation Metrics. We use three evaluation metrics: MAE, R^2, and SRCC, to assess the performance of semi-supervised regression methods. MAE measures the average absolute difference between the model's predictions and the actual values. The R^2 score indicates the proportion of variance in the data explained by the model. SRCC evaluates the correlation between the predicted rankings and the actual rankings.

Evaluation Robustness. To ensure the reliability of our evaluation results, each experiment is executed three times using fixed random seeds (0, 1, and 2). We report both the mean and standard deviation of each metric.

Fair Comparison. To ensure a fair comparison between our proposed methods and related works, we implement and evaluate all methods within the same codebase. Specifically, we adapt the popular semi-supervised classification framework USB [26], modifying it for regression tasks to implement both our proposed methods and related works. Weak augmentation is applied consistently to the labeled data across all semi-supervised and supervised methods. For specific details on the modifications made to USB, please refer to Appendix A.5. The code and full training logs of the experiments presented in this paper are open-sourced at `https://github.com/pm25/semi-supervised-regression`.

Hyperparameters. We use the hyperparameters of USB as the base for fine-tuning. We first fine-tune the hyperparameters in the supervised baseline setting and find the hyperparameters that lead to lowest MAE score. These same hyperparameters are then applied to all semi-supervised regression methods to ensure a fair comparison. Only the additional hyperparameters specific to each semi-supervised method are further tuned. For more details on the hyperparameters, please refer to Appendix A.13.

Base Model. The base model used in our experiments varies depending on the data type. For image data, we use Wide ResNet-28-2 [32], which is not pre-trained. For audio data, we use the pre-trained Whisper-base [20], and for text data, we use the pre-trained Bert-Small [12].

Dataset. To simulate the semi-supervised setting, we randomly sample a portion of the dataset as labeled data, treating the remainder as unlabeled. To evaluate performance, we use three diverse datasets: UTKFace [37], an image age estimation dataset; BVCC [8], an audio quality assessment dataset; and Yelp Review [1], a text sentiment analysis (opinion mining) dataset. For more detailed information about these datasets, please refer to Appendix A.11.

 https://doi.org/10.52202/079017-3413

Table 1: Comparison of RankUp with and without RDA against other methods on the UTKFace dataset, evaluated under two settings: 50 and 250 labeled samples, with the remaining images treated as unlabeled. The original UTKFace dataset comprises 18,964 training images.

	UTKFace (Image Age Estimation)					
	Labels = 50			Labels = 250		
	MAE↓	R^2↑	SRCC↑	MAE↓	R^2↑	SRCC↑
Supervised	14.13±0.56	0.090±0.092	0.371±0.071	9.42±0.16	0.540±0.014	0.712±0.010
Π-Model	13.82±1.02	0.100±0.086	0.387±0.092	9.45±0.30	0.534±0.030	0.706±0.015
Mean Teacher	13.92±0.20	0.127±0.037	0.423±0.023	8.85±0.25	0.586±0.020	0.745±0.013
CLSS	13.61±0.92	0.138±0.101	0.447±0.074	9.10±0.15	0.586±0.016	0.737±0.014
UCVME	13.49±0.95	0.157±0.110	0.412±0.127	8.63±0.17	0.626±0.006	0.767±0.007
MixMatch	11.44±0.45	0.401±0.028	0.674±0.035	7.95±0.15	0.692±0.013	0.832±0.008
RankUp (Ours)	9.96±0.62	0.514±0.043	0.703±0.019	7.06±0.11	0.751±0.011	0.835±0.008
RankUp + RDA (Ours)	**9.33**±0.54	**0.552**±0.041	**0.770**±0.009	**6.57**±0.18	**0.782**±0.012	**0.856**±0.005
Fully-Supervised	4.85±0.01	0.875±0.000	0.910±0.001	4.85±0.01	0.875±0.000	0.910±0.001

4.2 Main Results

To evaluate the performance of RankUp under different labeled data settings, we conducted experiments using the UTKFace dataset with 50 and 250 labeled samples. We tested two configurations of RankUp: one incorporating the RDA (RankUp + RDA) and the other without it (RankUp). Their performance was compared against other semi-supervised regression methods, with MixMatch specifically representing the consistency regularization component of the approach. Additionally, we included a supervised setting that used only the labeled data **without** incorporating any unlabeled data during training, as well as a fully-supervised setting that used **all** available data (both labeled and unlabeled), assuming the unlabeled data had known true labels. We also conducted experiments with a 2000 labeled samples setting; however, due to space limitations, the results for this configuration can be found in Appendix A.2.

The results are presented in Table 1. We observed that RankUp (without RDA) consistently outperforms existing semi-supervised regression methods, especially when the amount of labeled data is scarce. Specifically, in the 50-label setting, RankUp achieves at least a 12.9% improvement in MAE, a 28.2% improvement in R^2, and a 4.3% improvement in SRCC compared to other semi-supervised regression methods. In the 250-label setting, RankUp shows at least an 11.2% improvement in MAE, an 8.5% improvement in R^2, and a 0.4% improvement in SRCC.

Furthermore, the integration of RDA with RankUp further enhances the performance of RankUp. Specifically, in the 50-label setting, RankUp + RDA achieves an additional 6.3% improvement in MAE, a 7.4% improvement in R^2, and a 9.5% improvement in SRCC compared to RankUp alone. Similarly, in the 250-label setting, RankUp + RDA achieves an additional 6.9% improvement in MAE, a 4.1% improvement in R^2, and a 2.5% improvement in SRCC relative to RankUp.

These empirical results demonstrate the effectiveness of RankUp and RDA across different labeled settings. Another notable observation is that RankUp + RDA in the 50-label setting outperforms the supervised model that utilizes five times the labeled data (in the 250-label setting) across all three metrics. Specifically, RankUp + RDA achieves a 1.0% improvement in MAE, a 2.2% improvement in R^2, and an 8.1% improvement in SRCC while using only one-fifth of the labeled data, demonstrating its effectiveness in reducing labeling costs.

4.3 Additional Results on Audio and Text Data

To further assess the performance of RankUp across different data types and tasks, we evaluated it on the BVCC and Yelp Review datasets using 250 labeled samples. The results are presented in Table 2. The table demonstrates that RankUp also consistently outperforms existing semi-supervised regression methods on both audio and text datasets. Specifically, on the BVCC dataset, RankUp (without RDA) achieves at least a 5.6% improvement in MAE, a 6.3% improvement in R^2, and a 0.3% improvement in SRCC compared to other semi-supervised regression methods. On the Yelp

Table 2: Comparison of RankUp with and without RDA against other methods on the BVCC and Yelp Review datasets, evaluated under the 250-labeled samples setting. The BVCC dataset consists of 4,975 training audio samples, while the Yelp Review dataset contains 250,000 training text comments.

	BVCC (Audio Quality Assessment)			Yelp Review (NLP Opinion Mining)		
	Labels = 250			Labels = 250		
	MAE↓	R^2↑	SRCC↑	MAE↓	R^2↑	SRCC↑
Supervised	0.533±0.006	0.490±0.018	0.741±0.009	0.723±0.023	0.566±0.019	0.769±0.010
Π-Model	0.534±0.008	0.489±0.021	0.740±0.009	0.730±0.024	0.565±0.019	0.769±0.009
Mean Teacher	0.532±0.006	0.492±0.018	0.742±0.008	0.730±0.024	0.565±0.019	0.769±0.009
CLSS	0.499±0.010	0.534±0.027	0.748±0.008	0.721±0.010	0.543±0.011	0.748±0.002
UCVME	0.498±0.003	0.553±0.011	0.774±0.008	0.775±0.006	0.540±0.005	0.763±0.005
MixMatch	0.597±0.017	0.353±0.044	0.626±0.031	0.886±0.004	0.381±0.008	0.660±0.004
RankUp (Ours)	0.470±0.012	0.588±0.028	0.776±0.010	0.661±0.018	0.645±0.013	**0.829**±0.002
RankUp + RDA (Ours)	**0.463**±0.013	**0.598**±0.027	**0.783**±0.011	**0.632**±0.009	**0.651**±0.007	0.810±0.005
Fully-Supervised	0.351±0.003	0.764±0.002	0.874±0.001	0.418±0.003	0.799±0.002	0.896±0.001

Table 3: Comparison of using different semi-supervised classification methods for training RankUp's ARC component. Results are evaluated on the UTKFace dataset with a setting of 250 labeled samples.

	MAE↓	R^2↑	SRCC↑
None	9.42±0.16	0.540±0.014	0.712±0.010
Supervised	9.03±0.09	0.588±0.018	0.746±0.008
Π-Model	8.81±0.11	0.591±0.013	0.751±0.012
Mean Teacher	8.76±0.13	0.607±0.018	0.750±0.005
FixMatch	**7.06**±0.11	**0.751**±0.011	**0.835**±0.008

Review dataset, RankUp shows at least a 8.3% improvement in MAE, a 14.2% improvement in R^2, and a 7.8% improvement in SRCC relative to other semi-supervised regression methods.

Moreover, integrating RDA with RankUp further enhances RankUp's performance on both datasets. In the BVCC dataset, RankUp + RDA achieves an additional 1.5% improvement in MAE, a 1.7% improvement in R^2, and a 0.9% improvement in SRCC. In the Yelp Review dataset, RankUp + RDA shows an additional 4.4% improvement in MAE and a 0.9% improvement in R^2. However, in the Yelp Review dataset, RankUp + RDA did not improve SRCC compared to RankUp without RDA, resulting in a 2.3% decrease in SRCC. This decline is likely due to the limited distinct label values (only five distinct values) in the Yelp Review dataset; applying RDA may cause many pseudo-labels to align with the same value, thereby disrupting the ranking relationships and leading to a degradation in SRCC performance.

4.4 Analysis of Different Semi-Supervised Classification Methods on ARC

To better understand the effect of RankUp's ARC component and its ability to utilize the unlabeled data, we analyzed ARC by training it with various semi-supervised classification methods. In this evaluation, we did not apply RDA to isolate its effect on ARC. Additionally, we compared these methods against a baseline with no ARC (denoted as "None") and a supervised setting where only labeled data was used to train ARC (denoted as "Supervised").

The results are presented in Table 3. The table indicates that using an ARC without any unlabeled data can already improve performance, with a 4.1% improvement in MAE, an 8.9% improvement in R^2, and a 4.8% improvement in SRCC compared to the "None" setting. This suggests the beneficial impact of training ARC concurrently with the original regression head, even without leveraging unlabeled data.

https://doi.org/10.52202/079017-3413

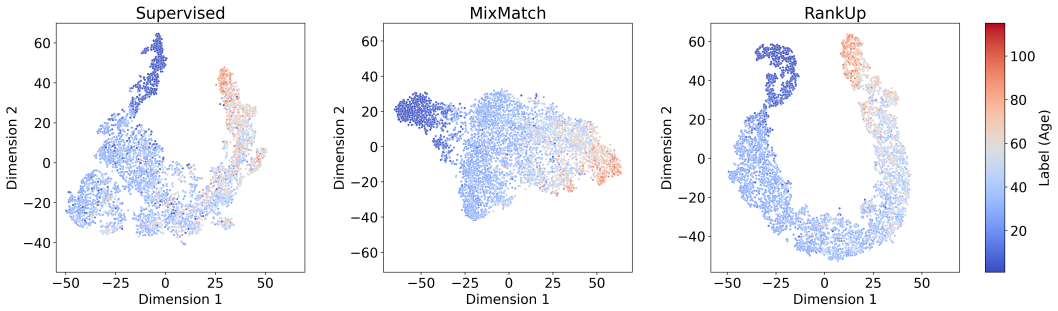

Figure 3: Comparison of t-SNE visualizations of feature representations for different semi-supervised regression methods on evaluation data. The supervised model is displayed on the left, MixMatch is in the center, and RankUp (without RDA) is shown on the right.

Furthermore, using different semi-supervised classification methods to utilize the unlabeled data further boost performance compared to using only labeled data. Specifically, we tested the Π-Model, Mean Teacher, and FixMatch, all of which demonstrated improvements over the "Supervised" setting across MAE, R^2, and SRCC metrics. Among these methods, FixMatch achieved the best results, showing a 21.8% improvement in MAE, a 27.7% improvement in R^2, and an 11.9% improvement in SRCC compared to the "Supervised" setting. This highlights the effectiveness of leveraging unlabeled data and semi-supervised classification techniques to improve ARC and RankUp's overall performance.

4.5 Understanding Smoothness and Low-Density Assumptions in Regression

The low-density assumption is crucial for understanding the effectiveness of semi-supervised learning methods. However, it does not directly apply to regression tasks due to the absence of decision boundaries and confidence measures. In this section, we explore why RankUp performs well in regression tasks by leveraging semi-supervised classification techniques that utilize the low-density assumption. This understanding can broaden our perspective on these assumption.

We adopt a broader interpretation of the smoothness and low-density assumptions. Rather than viewing them solely in the context of classification, we interpret the smoothness assumption as an effort to **group features with similar labels together**, while the low-density assumption aims to **separate features with dissimilar labels**. These perspectives align with RankUp's approach. By training ARC with pseudo-labels, the model is encouraged to have greater confidence in the pairwise ranking predictions of the unlabeled data, thus pushing the features with dissimilar pseudo-labels further apart. Additionally, ensuring consistent predictions between weakly and strongly augmented data assists in grouping features with similar labels. The t-SNE visualization demonstrated in Figure 3 supports this claim, showing that within RankUp, similar labels are closer together, while dissimilar labels are pushed further apart in the feature space.

5 Conclusion

Recent advancements in semi-supervised learning have achieved impressive results across various classification tasks; however, these methods are not directly applicable to regression tasks. In this work, we investigate the potential of leveraging existing semi-supervised classification techniques for regression tasks. We propose a novel framework, RankUp, which introduces two key components: the Auxiliary Ranking Classifier (ARC) and Regression Distribution Alignment (RDA). The empirical results of our experiments demonstrate the effectiveness of our methods across various labeled data settings (50, 250, and 2000 labeled samples) and different types of datasets (image, audio, and text). These findings show that semi-supervised classification techniques can be effectively adapted to regression tasks, bridging the gap between research in semi-supervised regression and classification, and paving the way for more advanced research in this area.

6 Acknowledgments

I sincerely appreciate everyone who made this work possible. I am especially thankful to my coauthors, Szu-Wei Fu and Prof. Yu Tsao for their mentorship and our weekly discussions; their insights and comments have been invaluable in shaping this research. I am also very grateful for my experience at CLLab and the guidance of Hsuan-Tien Lin, which sparked my interest in weakly-supervised learning and greatly influenced my research mindset and approach. My heartfelt thanks go to my family for their unwavering support throughout my academic journey. I also thank my friend Chi-Chang, whose discussions helped me develop a deeper understanding of how to approach research. Lastly, I want to thank Chia-Ling for her continuous support and encouragement during the development of this research. This work would not have been possible without all these individuals. Thank you!

References

[1] Yelp dataset: http://www.yelp.com/dataset_challenge.

[2] D. Berthelot, N. Carlini, E. D. Cubuk, A. Kurakin, K. Sohn, H. Zhang, and C. Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations*, 2020.

[3] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

[4] D. Berthelot, R. Roelofs, K. Sohn, N. Carlini, and A. Kurakin. Adamatch: A unified approach to semi-supervised learning and domain adaptation. In *International Conference on Learning Representations*, 2022.

[5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.

[6] O. Chapelle, B. Scholkopf, and A. Zien. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009.

[7] H. Chen, R. Tao, Y. Fan, Y. Wang, J. Wang, B. Schiele, X. Xie, B. Raj, and M. Savvides. Softmatch: Addressing the quantity-quality tradeoff in semi-supervised learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[8] E. Cooper and J. Yamagishi. How do voices from past speech synthesis challenges compare today? *arXiv preprint arXiv:2105.02373*, 2021.

[9] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[10] W. Dai, X. Li, and K.-T. Cheng. Semi-supervised deep regression with uncertainty consistency and variational model ensembling via bayesian neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7304–7313, 2023.

[11] W. Dai, D. Yao, H. Bai, K.-T. Cheng, and X. Li. Semi-supervised contrastive learning for deep regression with ordinal rankings from spectral seriation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

 https://doi.org/10.52202/079017-3413

[14] S. C. Hoi, R. Jin, J. Zhu, and M. R. Lyu. Batch mode active learning and its application to medical image classification. In *Proceedings of the 23rd international conference on Machine learning*, pages 417–424, 2006.

[15] V. Hosu, H. Lin, T. Sziranyi, and D. Saupe. Koniq-10k: An ecologically valid database for deep learning of blind image quality assessment. *IEEE Transactions on Image Processing*, 29:4041–4056, 2020.

[16] J. Kim, Y. Hur, S. Park, E. Yang, S. J. Hwang, and J. Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *Advances in neural information processing systems*, 33:14567–14579, 2020.

[17] S. Laine and T. Aila. Temporal ensembling for semi-supervised learning. In *International Conference on Learning Representations*, 2017.

[18] D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013.

[19] J. Lorenzo-Trueba, J. Yamagishi, T. Toda, D. Saito, F. Villavicencio, T. Kinnunen, and Z. Ling. The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods. In *The Speaker and Language Recognition Workshop (Odyssey 2018)*, page 195. ISCA, 2018.

[20] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023.

[21] S. Rahimi, O. Oktay, J. Alvarez-Valle, and S. Bharadwaj. Addressing the exorbitant cost of labeling medical images with active learning. In *International Conference on Machine Learning in Medical Imaging and Analysis*, volume 1, 2021.

[22] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

[23] R. G. D. Steel, J. H. Torrie, et al. Principles and procedures of statistics. *Principles and procedures of statistics.*, 1960.

[24] A. Tarvainen and H. Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

[25] J. E. Van Engelen and H. H. Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020.

[26] Y. Wang, H. Chen, Y. Fan, W. Sun, R. Tao, W. Hou, R. Wang, L. Yang, Z. Zhou, L.-Z. Guo, H. Qi, Z. Wu, Y.-F. Li, S. Nakamura, W. Ye, M. Savvides, B. Raj, T. Shinozaki, B. Schiele, J. Wang, X. Xie, and Y. Zhang. Usb: A unified semi-supervised learning benchmark for classification. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

[27] Y. Wang, H. Chen, Q. Heng, W. Hou, Y. Fan, , Z. Wu, J. Wang, M. Savvides, T. Shinozaki, B. Raj, B. Schiele, and X. Xie. Freematch: Self-adaptive thresholding for semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2023.

[28] C. Wei, K. Sohn, C. Mellina, A. Yuille, and F. Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10857–10866, 2021.

[29] C. J. Willmott and K. Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82, 2005.

[30] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le. Unsupervised data augmentation for consistency training. *Advances in neural information processing systems*, 33:6256–6268, 2020.

[31] Y. Xu, L. Shang, J. Ye, Q. Qian, Y.-F. Li, B. Sun, H. Li, and R. Jin. Dash: Semi-supervised learning with dynamic thresholding. In *International Conference on Machine Learning*, pages 11525–11536. PMLR, 2021.

[32] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[33] J. H. Zar. Significance testing of the spearman rank correlation coefficient. *Journal of the American Statistical Association*, 67(339):578–580, 1972.

[34] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021.

[35] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[36] W. Zhang, L. Zhu, J. Hallinan, S. Zhang, A. Makmur, Q. Cai, and B. C. Ooi. Boostmis: Boosting medical image semi-supervised learning with adaptive pseudo labeling and informative active annotation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20666–20676, 2022.

[37] Z. Zhang, Y. Song, and H. Qi. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5810–5818, 2017.

 https://doi.org/10.52202/079017-3413

A Appendix / supplemental material

A.1 Limitations

One of the limitations of RDA is that it relies on two key assumptions: the labeled and unlabeled data have similar label distributions, and the ranking of the pseudo-labels is accurate. If either of these assumptions is not met, the performance of RDA can be significantly impacted.

Another limitation of RDA is that it may not perform well in tasks where distinct label values are few. As shown in Table 2, in the Yelp Review dataset, SRCC decreases when RankUp is combined with RDA compared to using RankUp alone. This is likely because the limited number of distinct label values causes many pseudo-labels to align to the same value, disrupting the ranking relationships.

Despite these limitations, RDA is still a powerful component that can work with RankUp to further boost its performance. However, users should carefully evaluate their dataset characteristics to determine whether applying RDA is an appropriate choice.

A.2 Analysis on UTKFace 2000 Labeled Setting

In Table 1, we present the results of RankUp against other methods using 50 and 250 labeled samples settings on the UTKFace dataset. Here, we further investigate the performance of RankUp in a 2000-label setting, with the results shown in Table 4. Notably, RankUp continues to exhibit performance improvements over other semi-supervised regression methods in a 2000-label setting. Specifically, RankUp (without RDA) demonstrates a 7.0% improvement in MAE, a 1.7% improvement in R^2, and a 0.5% improvement in SRCC. Meanwhile, RankUp with RDA demonstrates an additional 1.8% improvement in MAE, a 0.7% improvement in R^2, and a 0.3% improvement in SRCC compared to RankUp alone. Comparing the results in Table 1 (50 and 250 labeled samples) with those in Table 4 (2000 labeled samples), we observe that RankUp with or without RDA continue to outperform other methods as the number of labeled samples increases. However, as expected, the advantages of semi-supervised learning diminish with the availability of more labeled training data.

Table 4: Comparison of RankUp with and without RDA against other methods on the UTKFace dataset with 2000 labeled samples.

	UTKFace (Image Age Estimation)		
	Labels = 2000		
	MAE↓	R^2↑	SRCC↑
Supervised	6.28±0.06	0.794±0.004	0.862±0.001
Π-Model	6.31±0.10	0.790±0.006	0.860±0.003
Mean Teacher	6.29±0.03	0.793±0.004	0.862±0.001
CLSS	6.29±0.01	0.794±0.003	0.862±0.001
UCVME	5.90±0.07	0.821±0.007	0.877±0.002
MixMatch	6.03±0.07	0.824±0.004	0.883±0.002
RankUp (Ours)	5.61±0.07	0.838±0.003	0.887±0.003
RankUp + RDA (Ours)	**5.51**±0.07	**0.844**±0.004	**0.890**±0.003
Fully-Supervised	4.85±0.01	0.875±0.000	0.910±0.001

A.3 Analysis of Different Semi-Supervised Regression Methods on RankUp

We evaluated the impact of different semi-supervised regression methods on training RankUp's regression output. The ARC was trained using FixMatch, while the regression output was trained with various semi-supervised regression techniques. We compared these results with a supervised setting, where only labeled data was used for training the regression output (denoted as "Supervised"). The results, shown in Table 5, indicate that using different semi-supervised regression methods can further improve performance in MAE, R^2, and SRCC metrics compared to the supervised method alone. Demonstrating the effectiveness of leveraging unlabeled data to directly train the regression head.

Among all the semi-supervised regression methods we tested, our proposed RDA achieves the best performance in terms of MAE and R^2, with at least a 7.7% improvement in MAE and a 1.7% improvement in R^2 compared to other methods. However, it shows a decrease in SRCC compared to MixMatch, with a 1.1% drop. In theory, RDA can also be used alongside the Π-Model, Mean Teacher, and MixMatch, as it focuses on refining pseudo-labels, which is distinct from the approaches of these methods. However, due to increased computational expense and the goal of maintaining a simpler framework, our proposed RankUp only uses a supervised method or RDA for training the regression output.

Table 5: Comparison of using different semi-supervised regression methods for training RankUp's regression output. Results are evaluated on UTKFace dataset with a setting of 250 labeled samples.

	MAE↓	R^2↑	SRCC↑
Supervised	7.06±0.11	0.751±0.011	0.835±0.008
Π-Model	6.95±0.16	0.758±0.010	0.837±0.005
Mean Teacher	7.01±0.17	0.752±0.013	0.831±0.004
MixMatch	7.12±0.09	0.769±0.006	**0.866**±0.002
RDA (Ours)	**6.57**±0.18	**0.782**±0.012	0.856±0.005

A.4 Ablation Studies on RankUp Components: ARC and RDA

To further understand the effect of ARC and RDA in RankUp, we conducted ablation studies comparing the performance of a supervised baseline, RDA only, ARC only, and ARC + RDA on the UTKFace dataset, using settings of 50 and 250 labeled samples.

The results, as shown in Table 6, demonstrate that RDA alone does not consistently outperform the supervised baseline, particularly in the 50 labeled samples setting (as reflected in the MAE and R^2 values). This may be due to one of the assumptions of RDA not being met, where it requires the pseudo-labels to have reasonably accurate ranking (the supervised baseline with 50 labeled samples only has an SRCC score of 0.371). In contrast, ARC alone ("RankUp" in the table) significantly improves performance over both the supervised baseline and RDA alone. The combination of ARC and RDA yields the best performance, highlighting their synergistic relationship, where it is beneficial to use RDA with ARC.

Table 6: Ablation studies on the ARC and RDA components introduced in RankUp. In this table, "RankUp" refers to the use of the ARC component only, while "RDA" refers to the use of the RDA component only. "RankUp + RDA" refers to the combined use of both ARC and RDA. Results are evaluated on the UTKFace dataset with 50 and 250 labeled samples.

	UTKFace (Image Age Estimation)					
	Labels = 50			Labels = 250		
	MAE ↓	R2 ↑	SRCC ↑	MAE ↓	R2 ↑	SRCC ↑
Supervised	14.13±0.56	0.090±0.092	0.371±0.071	9.42±0.16	0.540±0.014	0.712±0.010
RDA (Ours)	14.34±1.27	0.060±0.125	0.442±0.104	8.64±0.22	0.609±0.023	0.772±0.012
RankUp (Ours)	9.96±0.62	0.514±0.043	0.703±0.019	7.06±0.11	0.751±0.011	0.835±0.008
RankUp + RDA (Ours)	9.33±0.54	0.552±0.041	0.770±0.009	6.57±0.18	0.782±0.012	0.856±0.005

A.5 Modifications for Adapting USB Codebase for Regression Tasks

The USB [26] codebase is originally designed for semi-supervised classification tasks and includes implementations of various existing semi-supervised classification methods. To adapt it for regression tasks, enabling the implementation of our proposed framework RankUp and other semi-supervised regression methods, we made the following key modifications:

1. Replaced the cross entropy loss function with the MAE loss function.

 https://doi.org/10.52202/079017-3413

2. Adjusted the output layer to produce a single continuous output instead of multiple outputs used for multi-class classification.
3. Removed one-hot encoding from the codebase.
4. Normalized the regression labels to the 0-1 range.
5. For methods like Mean Teacher [24] and Π-Model [17], no changes were needed to the core algorithms, as they are inherently applicable to regression tasks.
6. For MixMatch [3], we excluded components specifically designed for classification, such as sharpening and one-hot label encoding, while retaining the input mixing and consistency regularization aspects, which are valuable for regression tasks.

A.6 Influence of Data Augmentation on Labeled Data

To further analyze the effect of weak augmentation on labeled data, we conducted experiments on the UTKFace dataset with a setting of 250 labeled samples, comparing the results with and without weak augmentation applied to the labeled data.

Table 7 presents a comparison of the results, illustrating the significance of weak augmentation. The performance metrics (MAE, R^2, SRCC) show a decline when weak augmentation is not applied (note that weak augmentation was applied to all the baselines in the paper). Despite the observed drop in performance when weak augmentation is not applied, the relative order of effectiveness among the methods remains consistent, regardless of the application of weak augmentation. This consistency further underscores the robustness of our proposed method.

Table 7: Analysis of the effect of weak augmentation on labeled data. Results are evaluated on the UTKFace dataset with a setting of 250 labeled samples. Best results for applying or not applying weak augmentation for each method are highlighted in bold.

	Weak Augmentation	MAE ↓	R2 ↑	SRCC ↑
Supervised	Yes	**9.42**±0.16	**0.540**±0.014	**0.712**±0.010
	No	11.73±0.17	0.315±0.012	0.556±0.020
Π-Model	Yes	**9.45**±0.30	**0.534**±0.030	**0.706**±0.015
	No	11.97±0.39	0.319±0.029	0.567±0.026
MixMatch	Yes	**7.95**±0.15	**0.692**±0.013	**0.832**±0.008
	No	10.80±0.09	0.446±0.021	0.716±0.003
RankUp + RDA (Ours)	Yes	**6.57**±0.18	**0.782**±0.012	**0.856**±0.005
	No	7.16±0.25	0.742±0.022	0.843±0.010
Fully-Supervised	Yes	**4.85**±0.01	**0.875**±0.000	**0.910**±0.001
	No	5.58±0.02	0.837±0.002	0.888±0.000

A.7 Data Augmentation Operators

We followed the settings in the USB [26] codebase for augmentation operators, with an adjustment to the strong augmentation for audio data. This adjustment was necessary because our task involves quality assessment, and the original strong augmentation method would have affected the quality of the data. Specifically, we used the following augmentation techniques:

Image

- Weak augmentation: Random Crop, Random Horizontal Flip
- Strong augmentation: RandAugment [9]

Audio

- Weak augmentation: Random Sub-sample
- Strong augmentation: Random Sub-sample, Random Mask, Random Trim, Random Padding

Text

- Weak augmentation: None
- Strong augmentation: Back-Translation [30]

A.8 Hardware Specifications

All experiments reported in this paper were conducted using an Nvidia Titan XP with 12 GB of VRAM and an Nvidia GeForce RTX 2080 Ti, also equipped with 12 GB of VRAM.

A.9 More Feature Visualization Results

Additional feature visualization results beyond those presented in this paper are available. This includes t-SNE and UMAP visualizations, in both 2D and 3D, for different semi-supervised regression methods, all accessible at `https://github.com/pm25/semi-supervised-regression`.

A.10 Dataset Processing

In our experiments, if the dataset provides a pre-defined train-eval-test split, we utilize the training split to train the model and evaluate its performance on the evaluation or test split. If the dataset does not provide such a split, we randomly sample 80% of the data as the training set and the remaining 20% as the test set. We open-source the train-test splits used for conducting the experiments in this paper at `https://github.com/pm25/regression-datasets`.

A.11 Datasets

Three datasets are utilized in the experiments conducted in this paper: UTKFace [37], BVCC [8], and the Yelp Review [1] dataset. Below, we provide a brief introduction to each dataset.

UTKFace. The UTKFace dataset is an image age estimation dataset, where the goal is to predict the age of the person in an image. The labels range from 1 to 116 years old. The dataset consists of 23,705 face images, which we split into 18,964 training samples and 4,741 test samples. The dataset is available in two versions: the original images and an aligned and cropped version. The experiments conducted in this paper use the aligned and cropped version of the UTKFace dataset.

BVCC. The VoiceMOS2022 (BVCC) dataset is an audio quality assessment dataset, where the objective is to predict the quality of an audio sample. The labels, ranging from 1 to 5, are obtained by averaging the scores provided by multiple listeners. The dataset is split into training (4,974 samples), evaluation (1,066 samples), and testing (1,066 samples) sets. In the experiments reported in this paper, we utilize only the training and evaluation splits to evaluate performance.

Yelp Review. The Yelp Review dataset is a text opinion mining task, where the goal is to predict the rating of customers based on the comments they leave on the Yelp website. There are only five distinct ratings (0 to 4). We use the processed Yelp Review data provided by the USB [26] codebase. The dataset comprises training (250,000 samples), evaluation (25,000 samples), and testing (10,000 samples) sets. We only utilize the training split for model training and the evaluation set for evaluation.

A.12 Hyperparameter Fine-Tuning

We began by fine-tuning the hyperparameters of the supervised baseline for each dataset, initially setting them based on the configurations provided in the USB [26] codebase. We then fine-tuned the learning rate, weight decay, and layer decay hyperparameters for the supervised baseline model to identify the optimal set of hyperparameters that produced the lowest MAE score. The same hyperparameter settings were subsequently applied to all other methods evaluated in this study. Additionally, each method underwent further fine-tuning by adjusting its own specific additional hyperparameters, distinct from those used in the supervised setting.

 https://doi.org/10.52202/079017-3413

A.13 Hyperparameters

In this section, we list the hyperparameters used in each experimental setting presented in the paper. Table 8 provides the common hyperparameters for the base models: Wide ResNet-28-2 (for image data), Whisper-Base (for audio data), and Bert-Small (for text data). Specific hyperparameter configurations for each semi-supervised regression method are detailed in Table 9. The full code and hyperparameters are open-sourced at `https://github.com/pm25/semi-supervised-regression`.

Table 8: Common hyperparameters for the base models: Wide ResNet-28-2 (for image data), Whisper-Base (for audio data), and Bert-Small (for text data).

	Wide ResNet-28-2	Whisper-Base	Bert-Small
Training Iterations	262,144	102,400	102,400
Evaluation Iterations	1,024	1,024	1,024
Training Batch Size	32	8	8
Optimizer	SGD	AdamW	AdamW
Momentum	0.9	-	-
Criterion	MAE	MAE	MAE
Weight Decay	1e-03	2e-05	5e-04
Layer Decay	1.0	0.75	0.75
Learning Rate	1e-02	2e-06	1e-05
EMA Weight	0.999	-	-
Pretrained	False	True	True
Sampler	Random	Random	Random
Image Resize	40x40	-	-
Max Length Seconds	-	6.0	-
Sample Rate	-	16,000	-
Max Length	-	-	512

Table 9: Specific hyperparameters for each semi-supervised regression methods.

	Π-Model, MeanTeacher	MixMatch	UCVME	CLSS	RankUp	RankUp +RDA
Unlabeled Batch Ratio	1.0	1.0	1.0	0.25	7.0	7.0
Regression Unlabeled Loss Ratio	0.1	0.1	0.05	-	-	1.0
Regression Unlabeled Loss Warmup	0.4	0.4	-	-	-	0.4
Mixup Alpha	-	0.5	-	-	-	-
Dropout Rate	-	-	0.05	-	-	-
Ensemble Number	-	-	5	-	-	-
CLSS Lambda	-	-	-	2.0	-	-
Labeled Contrastive Loss	-	-	-	1.0	-	-
Unlabeled Contrastive Loss	-	-	-	0.05	-	-
Unlabeled Rank Loss Ratio	-	-	-	0.01	-	-
ARC Unlabeled Loss Ratio	-	-	-	-	1.0	1.0
ARC Loss Ratio	-	-	-	-	0.2	0.2
Confidence Threshold	-	-	-	-	0.95	0.95
Temperature	-	-	-	-	0.5	0.5
RDA Refinement Steps	-	-	-	-	-	1,024

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claim aligns with our contribution, which is that existing semi-supervised classification methods can be successfully adapted for use in semi-supervised regression tasks. Our empirical results demonstrate the effectiveness of our proposed framework in achieving this adaptation.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations of our proposed method in Section A.1.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/079017-3413

Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have provided a detailed implementation of our proposed framework in Section 3, including the hyperparameters used in our experiments, which can be found in Appendix A.13. This information can be used to reproduce the results presented in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We have provided details about the datasets used in our experiments in Section A.11. These datasets can be easily downloaded from the internet. If a dataset did not come with predefined splits, we manually created the train-test splits. The splits we used are open-sourced at `https://github.com/pm25/regression-datasets`. The code used in the experiments is also open-sourced at `https://github.com/pm25/semi-supervised-regression`, and can be used to reproduce the results presented in the paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The training and testing details of our experiments are outlined in Appendix A.12.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All of our experimental results were obtained by running each experiment three times with fixed random seeds 0, 1, and 2. We have provided the mean and standard deviation for our experimental results.

https://doi.org/10.52202/079017-3413

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: For more detailed information about the computer resources used for conducting our experiments, please refer to Appendix A.8.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have thoroughly reviewed and ensured that we have met all the statement outlined in the ethical guidelines.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: There is no societal impact of the work performed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly cited all the code, data, and models used in this paper to ensure proper attribution and transparency.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

https://doi.org/10.52202/079017-3413

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The documentation, along with the code used in the paper, is open-sourced at `https://github.com/pm25/semi-supervised-regression`. It provides detailed instructions on how to use the code. The code is derived from the USB[26] codebase, and both our modifications and the original USB code are licensed under the MIT license, which allows for modification and redistribution. More details about the license can be found at `https://github.com/pm25/semi-supervised-regression/blob/main/LICENSE`.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-3413