Semi-Supervised Regression with Heteroscedastic Pseudo-Labels

Xueqing Sun[1], Renzhen Wang[1]*, Quanziang Wang[1], Yichen Wu[2,3], Xixi Jia[4], Deyu Meng[1,5]
[1] Xi'an Jiaotong University [2] City University of Hong Kong [3] Harvard University
[4] Xidian University [5] Pazhou Laboratory (Huangpu)
xqsun@stu.xjtu.edu.cn, {rzwang,dymeng}@xjtu.edu.cn
{quanziangwang, wuyichen.am97, hsijiaxidian}@gmail.com

Abstract

Pseudo-labeling is a commonly used paradigm in semi-supervised learning, yet its application to semi-supervised regression (SSR) remains relatively under-explored. Unlike classification, where pseudo-labels are discrete and confidence-based filtering is effective, SSR involves continuous outputs with heteroscedastic noise, making it challenging to assess pseudo-label reliability. As a result, naive pseudo-labeling can lead to error accumulation and overfitting to incorrect labels. To address this, we propose an uncertainty-aware pseudo-labeling framework that dynamically adjusts pseudo-label influence from a bi-level optimization perspective. By jointly minimizing empirical risk over all data and optimizing uncertainty estimates to enhance generalization on labeled data, our method effectively mitigates the impact of unreliable pseudo-labels. We provide theoretical insights and extensive experiments to validate our approach across various benchmark SSR datasets, and the results demonstrate superior robustness and performance compared to existing methods. Our code is available at https://github.com/sxq/Heteroscedastic-Pseudo-Labels.

1 Introduction

Deep learning has achieved superior performance on various scenarios, particularly when large amounts of labeled data are available [28]. However, acquiring large-scale datasets with accurate labels is often costly or impractical in many real-world scenarios, such as medical imaging where labeled data is typically scarce and expensive or video analysis that requires labor-intensive frame-by-frame annotations [10, 51]. To address this challenge, Semi-Supervised Learning (SSL) has emerged as a potent paradigm that leverages vast amounts of unlabeled data to enhance learning performance, thereby reducing the dependency on expensive labeled annotations [44, 52].

In classification tasks, SSL techniques such as pseudo-labeling [29, 41, 50, 57] and consistency regularization [40, 43, 2, 1, 49] have been widely adopted, achieving impressive results across various domains. However, semi-supervised regression (SSR) presents unique challenges distinct from its classification counterpart. Unlike classification, where pseudo-labels are discrete and can be sharpened to encourage high-confidence predictions, regression inherently deals with continuous outputs, making it difficult to establish a reliable pseudo-labeling mechanism [18]. Furthermore, the lack of well-defined decision boundaries in regression complicates uncertainty quantification, increasing the risk of propagating incorrect pseudo-labels.

Due to these challenges, the pseudo-labeling methods commonly employed in semi-supervised classification are difficult to directly extend to semi-supervised regression due to the continuous

*Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

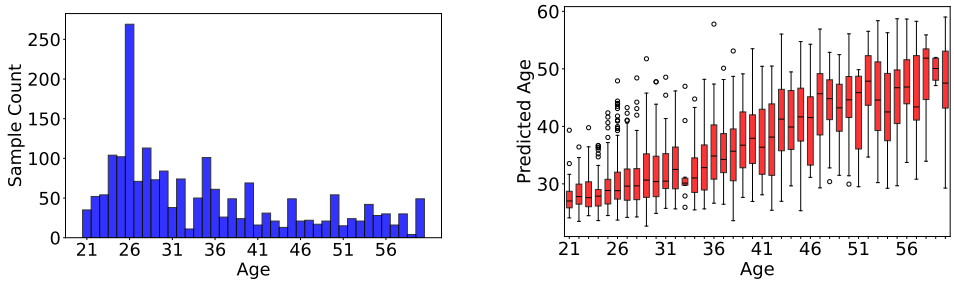


Figure 1: Experiment of UCVME [10] on UTKFace [58] with 10% labeled data. **Left:** Histogram of true labels for the unlabeled data. **Right:** Box-plot of pseudo-labels generated by UCVME.

nature of regression outputs. Consequently, existing SSR approaches focus primarily on leveraging consistency regularization to improve the utilization of unlabeled data. For example, TNNR [47] introduces a loop consistency that enforces consistency on the prediction differences between pairs of inputs. UCVME [10] proposes a novel uncertainty consistency loss for co-trained models to better account for model uncertainty. RankUp [18] reformulates the regression task as a ranking problem and imposes consistency of the rank between sample pairs.

These methods aim to enforce smoothness between predictions for labeled and unlabeled data, ensuring that the model learns a coherent mapping from input features to continuous outputs. However, the reliance on consistency regularization alone does not fully address the difficulties in handling uncertainty and the risk of overfitting to potentially incorrect pseudo-labels. To investigate this, we visualize the distribution of estimated pseudo-labels predicted by UCVME on the UTKFace dataset [58] with 10% labeled data. As shown in Fig. 1, despite the application of uncertainty consistency constraints, UCVME shows significant variance in pseudo-labels (right subfigure), even for age groups with large sample sizes (left subfigure). To address the aforementioned challenge, this paper introduces an uncertainty-aware pseudo-labeling method that dynamically adjusts the influence of pseudo-labeled samples based on calibrated uncertainty.

Our core motivation is that the uncertainty or noise in pseudo-labels varies with input features, leading to heteroscedasticity. Therefore, *it is crucial to develop an effective strategy for assigning appropriate uncertainty values to pseudo-labels, reflecting their varying degrees of error during training.* To achieve this, we propose a bi-level learning framework that not only minimizes empirical risk over both labeled and unlabeled data but also explicitly optimizes uncertainty estimates to improve generalization on labeled data during training. We theoretically and experimentally demonstrate that our method significantly enhances pseudo-label robustness and overall performance across various benchmark SSR tasks. In summary, our main contributions are: **(1)** We propose a bi-level SSR framework that explicitly optimizes the uncertainty of pseudo-labeled samples to improve the regression model's generalization when learning from potentially incorrect pseudo-labels. **(2)** We offer a theoretical analysis of the proposed method through gradient alignment, providing insights into why it enables the regression model to effectively mitigate the risk of overfitting to incorrect pseudo-labels. **(3)** We empirically validate our approach on various benchmark datasets and SSR settings, demonstrating its ability to improve both performance and robustness in SSR tasks.

2 Related work

2.1 Semi-Supervised Learning

Semi-Supervised Learning (SSL) [44, 52] trains models with both labeled and unlabeled data, which significantly reduces the cost of data annotation in real-world applications. SSL methods have been applied on wild scenarios, such as image classification [25, 41, 2], segmentation [59, 6, 16], and other tasks [54, 51, 33]. Current works on SSL can be categorized as follows: 1) *Pseudo-labeling methods*: These works [29, 41, 50, 57, 46, 17] train the model on both labeled data and unlabeled data with pseudo-labels, which are generated by the model itself. 2) *Consistency-based methods*: These models [40, 43, 2, 1, 49] apply prediction invariance loss on unlabeled data across perturbations. 3) *Generative methods*: These methods [42, 11, 4, 30] aim to model the real data distribution from the labeled and unlabeled data and generate new samples.

2.2 Semi-Supervised Regression

Semi-Supervised Regression (SSR), has attracted significant research interest in recent years, which includes 1) *Consistency-based methods*: Similar to the aforementioned methods in SSL, these methods designed different constraints to ensure prediction consistency in SSR. On the one hand, methods developed for semi-supervised classification can be readily extended to SSR, such as Mean Teacher [43] and Temporal Ensembling [26]. On the other hand, methods specifically designed for SSR have emerged. For example, TNNR [47] enforces loop consistency on prediction differences, UCVME [10] introduces an uncertainty consistency loss for co-trained models, and RankUp [18] enforces rank consistency between sample pairs. Additionally, CLSS [9] employs contrastive learning to encourage sample features with the same labels to be closer. 2) *Uncertainty-based methods*: In SSR, uncertainty estimation provides a potential way to improve the quality of pseudo-labels for unlabeled data. Specifically, SSDKL [20] proposes to minimize the prediction variance of the posterior regularization, and SimRegMatch [21] employs dropout to compute multiple predictions and use their variance to quantify the uncertainty of pseudo-labels. In a related manner, PIM [7] performs weakly supervised regression by weighting candidate labels according to their incurred losses, giving higher weights to labels with smaller losses. Unlike these methods, we propose an uncertainty-aware pseudo-labeling method from a bi-level optimization perspective in this work.

2.3 Uncertainty Estimation

Uncertainty estimation has been explored extensively in fully-supervised learning [27, 22, 19]. From a Bayesian viewpoint, Gal and Ghahramani [13] reinterpret Monte Carlo Dropout as variational inference for epistemic uncertainty estimation, which has been extended to semi-supervised classification methods [36] and semi-supervised segmentation tasks [56, 48] to filter unreliable pseudo-labels. Additionally, some works [55, 31] quantified uncertainty through prediction discrepancies between co-trained models. In the SSR task, UCVME [10], SSDKL [20], and SimRegMatch [21] have employed the uncertainty estimation methodology to improve the accuracy of pseudo-labels. Concretely, UCVME models pseudo-label noise under the heteroscedastic assumption, using a shared encoder with dual heads to predict both the mean and variance of the noise distribution. SSDKL combines a neural network encoder with a Gaussian Process in the latent space, enabling posterior inference to obtain predictive variance for unlabeled samples. SimRegMatch estimates the uncertainty of pseudo-labels by performing multiple forward passes with dropout, and selects lower uncertainty pseudo-labels through a thresholding strategy. Different from these methods, we propose a bi-level optimization framework and estimate pseudo-label uncertainty guided by labeled samples.

3 Method

3.1 Notations and Problem Setups

Semi-supervised regression (SSR) involves a labeled dataset $\mathcal{D}_l = \{(x_i^l, y_i^l)\}_{i=1}^N$ and an unlabeled dataset $\mathcal{D}_u = \{x_j^u\}_{j=1}^M$, where $x_i^l \in \mathcal{X}$ and $x_j^u \in \mathcal{X}$ represent training examples from the input space $\mathcal{X}$, and $y_i \in \mathbb{R}$ denotes the label associated with the labeled example x_i^l. The objective of SSR is to train a model f_θ that generalizes well on the test dataset.

In SSR, a widely used strategy involves employing pseudo-labeling techniques to augment the training dataset by assigning pseudo-labels to unlabeled data. In this strategy, each unlabeled example is assigned a pseudo-label based on the predictions of the model itself or its variants. The model is then trained on both labeled and unlabeled examples by optimizing an objective function $\mathcal{L} = \mathcal{L}_l + \lambda \mathcal{L}_u$. The loss function consists of two terms: the supervised loss $\mathcal{L}_l$ calculated on the labeled data, and the unsupervised loss $\mathcal{L}_u$ calculated on the pseudo-labeled data. The hyper-parameter λ balances the contributions between the supervised and unsupervised losses.

One of the key challenges in pseudo-labeling-based SSR is to effectively leverage pseudo-labeled data while mitigating the negative impact of incorrect pseudo-labels on model training. In this work, we propose a bi-level learning framework, which explicitly learns pseudo-label uncertainty to enhance the reliability and utility of pseudo-labeled data. By modeling the uncertainty associated with pseudo-labels, the method enables more robust training in SSR settings. In the following section, we first

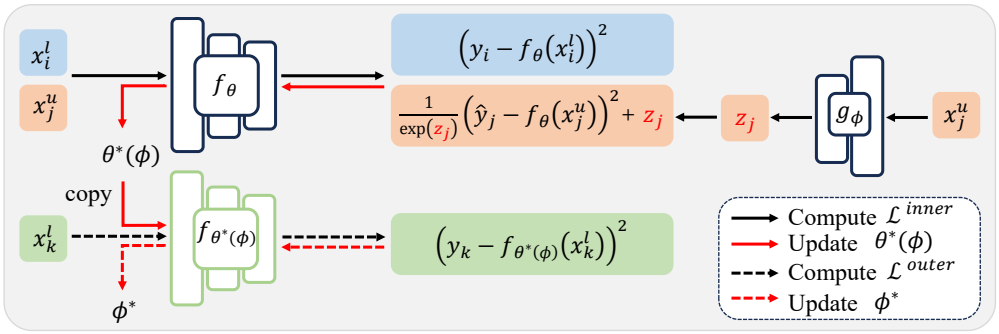


Figure 2: **Method Overview.** The proposed bi-level optimization framework consists of two main steps: (1) Inner-loop update, which updates the regression model using $\mathcal{L}^{inner}$ as defined in eq. (5), where $z_j = \log \sigma_j^2$; (2) Outer-loop update, which updates the uncertainty-learner using $\mathcal{L}^{outer}$ as defined in eq. (6). Note that we assume a batch size of 1 for better visualization.

introduce a probabilistic formulation to model the uncertainty in pseudo-labeling. We then describe how to effectively learn and incorporate pseudo-label uncertainty within our bi-level framework.

3.2 Heteroscedastic Pseudo-Labels

The objective of SSR is to learn the true label function $f_{\theta^*}(x)$ using both labeled and unlabeled data. However, when applying pseudo-labeling to SSR, the pseudo-labels assigned to the unlabeled data are usually imperfect. Moreover, the level of uncertainty or noise in these pseudo-labels varies across samples, particularly when the label estimation process involves an external source, such as weak supervision, self-training, or a noisy teacher model.

To model this relationship, we treat the pseudo-labels for each sample as heteroscedastic. Especially, for each unlabeled sample x_j^u and its corresponding pseudo-label $\hat{y}_j$, we assume that the variance of each pseudo-label depends on the input sample itself [3]. This can be formulated as:

$$\hat{y}_j = f_\theta(x_j^u) + \epsilon_j, \quad \epsilon_j \sim \mathcal{N}(0, \sigma_j^2) \tag{1}$$

where $f_\theta(x_j^u)$ is the true label prediction for input x_j^u, and ϵ_j is a noise term whose variance σ_j^2 varies with the input x_j^u. In *maximum likelihood* inference, we can write the negative log-likelihood (NLL) of the observation variable $\hat{y}_j$ as:

$$-\log p(\hat{y}_j \,|\, x_j^u) \propto \frac{(\hat{y}_j - f_\theta(x_j^u))^2}{\sigma_j^2} + \log(\sigma_j^2). \tag{2}$$

As such, for a batch of unlabeled data $\mathcal{B}_u$, we can define the unsupervised loss as the sum of the negative log-likelihood over all these samples:

$$\mathcal{L}_u = \sum_{x_j^u \in \mathcal{B}_u} \frac{1}{\sigma_j^2}(\hat{y}_j - f_\theta(x_j^u))^2 + \sum_{x_j^u \in \mathcal{B}_u} \log(\sigma_j^2). \tag{3}$$

Notably, if all pseudo-labels are assumed to be homoscedastic and σ is fixed at 1, the above loss degrades to the standard mean squared error (MSE) loss, which is commonly employed in various SSR methods.

In eq. (3), when an incorrect pseudo-label $\hat{y}_j$ is assigned to x_j^u during training, the model can mitigate the issue of error accumulation in f_θ by increasing the uncertainty of the pseudo-label, i.e., adjusting σ_j to downweight its contribution to the overall loss. In contrast, the standard MSE loss enforces rigid fitting of incorrect pseudo-labels, thereby exacerbating confirmation bias as training progresses. *Thus, a key challenge is to devise an effective strategy for dynamically assigning appropriate uncertainty values to pseudo-labels with varying degrees of error during training.*

3.3 Learning Uncertainty via Bi-level Optimization

To dynamically assign uncertainty values to pseudo-labels, a natural strategy is to introduce an auxiliary network g_ϕ parameterized by ϕ to produce σ_j for each unlabeled sample x_j^u, and then

jointly optimize the parameters $\{\theta, \phi\}$ of f_θ and g_ϕ in an end-to-end manner. However, this strategy suffers from a fundamental limitation, as it fails to distinguish between two distinct scenarios:

- Hard but correct samples: The pseudo-label $\hat{y}_j$ is correct, but the model prediction $f_\theta(x_j^u)$ is inaccurate, leading to a large squared error $(\hat{y}_j - f_\theta(x_j^u))^2$.
- Easy but incorrect samples: The pseudo-label $\hat{y}_j$ is incorrect, but the model prediction $f_\theta(x_j^u)$ is already close to the true label, also resulting in a large squared error.

The two scenarios result in a large squared error in the numerator of the first term in eq. (3), leading to a large σ_j^2 during minimization. However, the increased σ_j^2 would suppress the learning for hard but correct samples, which is crucial to improve the performance of SSR models.

Ideally, we seek a formulation where σ_j selectively suppresses unreliable pseudo-labels while still enforcing learning on difficult but valid samples. To this end, we propose a novel bi-level optimization framework that introduces a supportive model, referred to as the *uncertainty-learner*, to learn the uncertainty associated with pseudo-labels. Specifically, we parameterize the uncertainty-learner as a lightweight network g_ϕ that maps each pseudo-labeled sample x_j^u (or certain conditional information derived from the regression model, as elaborated in the next subsection) to its corresponding log-variance. This mapping can be formally expressed as:

$$z_j := \log \sigma_j^2 = g_\phi(x_j^u), \tag{4}$$

where ϕ denotes the parameters of g_ϕ. Note that we predict the log-variance rather than the variance directly, as this approach enhances numerical stability. Directly predicting σ_j^2 can lead to instability due to potential division by zero in the loss function [23, 22].

Our proposed bi-level optimization framework, which jointly optimizes the regression network f_θ and uncertainty-learner g_ϕ, is illustrated in Fig. 2 and can be described in the following two steps:

- **Inner-loop for optimizing f_θ:** The regression network f_θ is trained on both labeled and pseudo-labeled data by considering the uncertainty of pseudo-labels as follows:

$$\theta^*(\phi) = \arg\min_\theta \mathcal{L}^{inner} := \mathcal{L}_l(\theta) + \lambda \mathcal{L}_u(\theta, \phi), \tag{5}$$

  where $\mathcal{L}_l(\theta) = \sum_{x_i^l \in \mathcal{B}_l} (y_i - f_\theta(x_i^l))^2$ represents the supervised loss computed over a batch of labeled data, and $\mathcal{L}_u(\theta, \phi) = \sum_{x_j^u \in \mathcal{B}_u} \frac{1}{\exp(z_j)} (\hat{y}_j - f_\theta(x_j^u))^2 + \sum_{x_j^u \in \mathcal{B}_u} z_j$ denotes the unsupervised loss computed over a batch of pseudo-labeled data, with z_j obtained from eq. (4). The hyper-parameter λ controls the trade-off between $\mathcal{L}_l$ and $\mathcal{L}_u$. In eq. (5), ϕ essentially acts as hyper-parameters for θ, and in this step, we just express θ^* as a function of ϕ.
- **Outer-loop for optimizing g_ϕ:** Our ultimate objective is to generate well-calibrated uncertainty estimates that protect the regression model from incorrect pseudo-labels while ensuring good generalization on labeled data. To achieve this, we optimize ϕ by continuously tracking the performance of the updated model $f_{\theta^*(\phi)}$ on labeled data, which can be formulated as:

$$\phi^* = \arg\min_\phi \mathcal{L}^{outer} := \sum_{x_k^l \in \hat{\mathcal{B}}_l} (y_k - f_{\theta^*(\phi)}(x_k^l))^2, \tag{6}$$

  where $\hat{\mathcal{B}}_l$ is another batch of labeled data from $\mathcal{D}_l$ that differs from $\mathcal{B}_l$ used in eq. (5), i.e., $\hat{\mathcal{B}}_l \neq \mathcal{B}_l$. This formulation aims to find the optimal hyper-parameter ϕ^* such that the regression model, obtained by optimizing eq. (5), should also have good performance over any unbiased and reliable training batch.

In fact, eq. (5) and (6) formulate a bi-level learning framework. In the inner loop eq. (5), the regression model f_θ is updated using both labeled data and pseudo-labeled data, with the latter calibrated based on the uncertainty estimates generated by g_ϕ. In the outer loop eq. (6), the parameters of g_ϕ are optimized to ensure that the refined model $f_{\theta^*(\phi)}$ achieves more reliable and robust performance.

3.4 Practical Implementations

Uncertainty-learner's architecture. In practice, the uncertainty-learner g_ϕ does not directly map each unlabeled sample to its log-variance z_j, as presented in eq. (4). Instead, to improve computational

efficiency, it takes the prediction of the regression model $r_j = f_\theta(x_j^u)$ and the corresponding pseudo-label $\hat{y}_j$ as its input. This mapping can be formally formulated as: $z_j = g_\phi(r_j, \hat{y}_j)$. The architecture of g_ϕ is implemented as a multi-layer perceptron (MLP) with one single hidden layer. Despite its simplicity, according to the universal approximation theorem, such a network is theoretically capable of approximating any continuous function defined on a compact set with arbitrary precision [14]. Note that although both r_j and $\hat{y}_j$ are derived from the regression model, they are not strictly identical in this work. This discrepancy arises from the use of different random augmentations during the training process. Inspired by FixMatch [41], we generate the pseudo-label $\hat{y}_j$ from a weakly augmented version of x_j^u, while r_j is obtained from a strongly augmented version of the same input.

Algorithm 1 Mini-batch Training Algorithm of the Method

Input: labeled / unlabeled data $\mathcal{D}_l$ / $\mathcal{D}_u$, labeled / unlabeled batch size n/m, max iterations T
Output: regression network parameter θ^*
Initialize the parameters θ^0 of the regression network and those of the uncertainty-learner ϕ^0
for $t = 0$ **to** T **do**
  $B_l = \{(x_i^l, y_i^l)\}_{i=1}^n \leftarrow \mathrm{GetBatch}(\mathcal{D}_l, n)$.
  $B_u = \{x_j^u\}_{j=1}^m \leftarrow \mathrm{GetBatch}(\mathcal{D}_u, m)$.
  $\hat{B}_l = \{(x_k^l, y_k^l)\}_{k=1}^n \leftarrow \mathrm{GetBatch}(\mathcal{D}_l, n)$.
  Compute inner loss $\mathcal{L}^{inner}$ by eq. (5).
  Update regression network θ^{t+1} by eq. (7).
  Compute outer loss $\mathcal{L}^{outer}$ by eq. (6).
  Update uncertainty-learner by ϕ^{t+1} by eq. (8).
end for

Training Algorithm. The objective function of the regression network f_θ in eq. (5) and that of the uncertainty-learner g_ϕ in eq. (6) formulate a bi-level optimization problem, where the solution of $\theta^*(\phi)$ and ϕ^* are nested with each other. Approximately, we update θ and ϕ in a gradient-based method following [32, 12, 35].

(1) **Update θ.** At iteration t, we fix the uncertainty-learner parameter ϕ^t and conduct one-step gradient descent w.r.t regression network parameter θ^t in the inner optimization problem eq. (5) as follows:

$$\theta^{t+1}(\phi^t) = \theta^t - \alpha \cdot \nabla_\theta \mathcal{L}^{inner}(\theta^t, \phi^t), \tag{7}$$

where $\alpha > 0$ is the learning rate of the regression network.

(2) **Update ϕ.** Based on the updated regression network parameter $\theta^{t+1}(\phi)$, we optimize ϕ^t in the outer optimization problem eq. (6) by gradient descent:

$$\phi^{t+1} = \phi^t - \beta \cdot \nabla_\phi \mathcal{L}^{outer}(\theta^{t+1}(\phi^t)), \tag{8}$$

where $\beta > 0$ is the learning rate of the uncertainty-learner.

Since $\nabla_\phi \mathcal{L}^{outer}$ in eq. (8) introduces a second-order derivative, it requires unrolling the second-order differentiation of the entire regression network. This process is computationally expensive and inefficient for deep models. To alleviate this computational burden, we introduce an efficient approximation algorithm. Specifically, we assume that ϕ is only correlated with the parameters of the regression head (i.e., a single fully connected layer) of f_θ. This allows us to only unroll the second-order derivation of the regression head in eq. (8). Since the regression head contains far fewer parameters than the entire regression network, our proposed algorithm is considerably more efficient than conventional bi-level optimization algorithms [32, 12, 35].

3.5 Theoretical Analysis

We herein theoretically analyze the proposed bi-level optimization framework and reveal how our method helps the regression model mitigate the risk of overfitting to incorrect pseudo-labels.

Theorem 1. *Let $\nabla_\theta \mathcal{L}^{inner}(\theta, \phi)$ and $\nabla_\theta \mathcal{L}^{outer}(\theta)$ be the gradients of the inner and outer loss w.r.t. θ, respectively. The bi-level optimization in eq.* (5) *and eq.* (6) *is equivalent to the following optimization problem:*

$$\min_\phi - \left\langle \nabla_\theta \mathcal{L}^{inner}(\theta, \phi), \nabla_\theta \mathcal{L}^{outer}(\theta) \right\rangle. \tag{9}$$

The proof is presented in Appendix A.1. This theorem demonstrates that optimizing the parameter ϕ in our bi-level framework fundamentally aligns the gradients of the inner and outer losses w.r.t. θ. Based on eq. (5) and eq. (6), the gradient-matching formulation in eq. (9) suggests that our method implicitly regularizes the gradients from labeled and unlabeled data under the guidance of the outer data, and the discrepancy between these gradients is modulated by ϕ through its influence on the inner loss. Specifically, inaccurate uncertainty estimated by g_ϕ would interfere with the inner loss

Table 1: Main results on UTKFace. The performance is reported in the form of 'mean ± std' across six random runs. **Bold** means the best results and our method are shown in gray cells.

Method	$\gamma = 5\%$		$\gamma = 10\%$		$\gamma = 20\%$	
	MAE ↓	R^2 ↑	MAE ↓	R^2 ↑	MAE ↓	R^2 ↑
Fully-Supervised	4.713 ± 0.039	0.652 ± 0.006	4.713 ± 0.039	0.652 ± 0.006	4.713 ± 0.039	0.652 ± 0.006
Supervised	6.135 ± 0.046	0.454 ± 0.008	5.618 ± 0.026	0.533 ± 0.005	5.278 ± 0.048	0.581 ± 0.005
Mean Teacher [43]	5.912 ± 0.071	0.476 ± 0.011	5.474 ± 0.052	0.537 ± 0.009	5.126 ± 0.040	0.584 ± 0.007
Temporal Ensembling [26]	5.963 ± 0.060	0.478 ± 0.008	5.437 ± 0.022	0.557 ± 0.001	5.123 ± 0.032	0.604 ± 0.002
SSDKL [20]	5.855 ± 0.080	0.469 ± 0.010	5.391 ± 0.040	0.551 ± 0.007	5.173 ± 0.039	0.589 ± 0.007
TNNR [47]	5.817 ± 0.057	0.505 ± 0.008	5.372 ± 0.052	0.563 ± 0.007	5.104 ± 0.025	0.602 ± 0.005
SimRegMatch [21]	5.794 ± 0.073	0.512 ± 0.020	5.338 ± 0.065	0.584 ± 0.009	5.164 ± 0.108	0.612 ± 0.011
UCVME [10]	5.862 ± 0.036	0.495 ± 0.006	5.227 ± 0.045	0.585 ± 0.006	**4.870 ± 0.038**	**0.634 ± 0.005**
CLSS [9]	6.063 ± 0.145	0.451 ± 0.018	5.621 ± 0.077	0.514 ± 0.013	5.429 ± 0.078	0.546 ± 0.011
RankUp [18]	5.719 ± 0.112	0.495 ± 0.012	5.252 ± 0.040	0.576 ± 0.004	4.914 ± 0.015	0.621 ± 0.002
Ours	**5.639 ± 0.035**	**0.523 ± 0.009**	**5.143 ± 0.075**	**0.597 ± 0.009**	4.929 ± 0.038	0.624 ± 0.004

gradient w.r.t. θ, thus increasing the empirical risk in eq. (9). In contrast, when g_ϕ estimates accurate uncertainty in the inner loss, it helps the regression model perform gradient decent in an appropriate direction consistent with the outer data, ensuring stable and effective semi-supervised learning.

4 Experiments

In this section, we experiment with three benchmarks to evaluate our method. The experimental setups are described in Section 4.1, and the main results under varying label ratios are presented in Section 4.2. Section 4.3 provides an ablation study to assess the contribution of each key component, along with further discussion to gain a deeper understanding of the proposed method.

4.1 Experimental Setups

Datasets. We evaluate the effectiveness of our algorithm on three benchmark datasets: UTKFace [58], an image-based age estimation dataset; IMDB-WIKI [39], a large-scale dataset for age estimation; and STS-B [5, 45], a benchmark for assessing semantic similarity between sentence pairs. Please refer to Appendix B.1 for more detailed information about these datasets. These datasets are commonly used in supervised and semi-supervised regression tasks [10, 18, 53, 34]. In this work, we conduct experiments with varying label ratios γ, where 5%, 10%, and 20% of the training data are labeled, and the remaining data are used as unlabeled samples.

Evaluation Metrics. For the image datasets UTKFace and IMDB-WIKI, we follow UCVME [10] in using Mean Absolute Error (MAE, ↓) and the coefficient of determination (R^2, ↑) for evaluation. For the text dataset STS-B, we adopt Mean Squared Error (MSE, ↓) and R^2, following DIR [53]. Note that ↓ and ↑ indicate that lower and higher values are better, respectively. Detailed definitions of these metrics are provided in Appendix B.2. All experiments are conducted six times using fixed random seeds (0 to 5), and we report the mean and standard deviation for each metric.

Training Details. We conduct all experiments within a unified codebase to ensure fair comparisons. Specifically, following [10], we use ResNet-50 [15] pretrained on ImageNet as the backbone for the UTKFace and IMDB-WIKI datasets. For the STS-B dataset, we adopt a BiLSTM-based regression model with GloVe word embeddings, in line with [45, 53]. For more details on the implementation details, please refer to Appendix B.3.

Comparison Methods. We compare our method with various state-of-the-art SSR methods, which can be roughly divided into two categories. 1) *Consistency-based methods*: These methods leverage consistency constraints in model predictions, including some adapted from classification methods such as Mean Teacher [43] and Temporal Ensembling [26], as well as methods specifically designed for SSR, including TNNR [47], UCVME [10], CLSS [10] and RankUp [18]. 2) *Uncertainty-based methods*: These approaches explicitly model uncertainty in data or predictions, including SSDKL [20] and SimRegMatch [21]. Please refer to Appendix B.4 for more details about these methods. Moreover, we construct two benchmarks to validate the effectiveness of SSR methods: 1) *Supervised*: A lower bound for SSR methods that is trained on the labeled data without incorporating any unlabeled data. 2) *Fully-Supervised*: An upper bound trained on all labeled and unlabeled data while assuming ground truth labels of unlabeled data are known.

Table 2: Main results on IMDB-WIKI. The performance is reported in the form of 'mean ± std' across six random runs. **Bold** means the best results and our method are shown in gray cells.

Method	$\gamma = 5\%$		$\gamma = 10\%$		$\gamma = 20\%$	
	MAE ↓	R^2 ↑	MAE ↓	R^2 ↑	MAE ↓	R^2 ↑
Fully-Supervised	7.974 ± 0.043	0.724 ± 0.002	7.974 ± 0.043	0.724 ± 0.002	7.974 ± 0.043	0.724 ± 0.002
Supervised	10.172 ± 0.077	0.610 ± 0.004	9.248 ± 0.052	0.657 ± 0.002	8.647 ± 0.099	0.690 ± 0.005
Mean Teacher [43]	9.492 ± 0.051	0.647 ± 0.002	8.633 ± 0.093	0.689 ± 0.002	8.191 ± 0.066	0.711 ± 0.002
Temporal Ensembling [26]	11.335 ± 0.114	0.532 ± 0.007	9.517 ± 0.064	0.639 ± 0.004	9.577 ± 0.126	0.639 ± 0.007
SSDKL [20]	10.116 ± 0.073	0.611 ± 0.004	9.488 ± 0.031	0.641 ± 0.002	9.056 ± 0.043	0.656 ± 0.003
TNNR [47]	10.069 ± 0.088	0.612 ± 0.005	9.309 ± 0.052	0.654 ± 0.003	8.640 ± 0.033	0.688 ± 0.001
SimRegMatch [21]	9.908 ± 0.097	0.628 ± 0.004	9.110 ± 0.166	0.665 ± 0.007	8.587 ± 0.094	0.693 ± 0.006
UCVME [10]	9.730 ± 0.156	0.633 ± 0.007	8.920 ± 0.039	0.673 ± 0.004	8.309 ± 0.117	0.698 ± 0.003
CLSS [9]	9.906 ± 0.058	0.621 ± 0.007	9.251 ± 0.107	0.656 ± 0.006	8.781 ± 0.070	0.681 ± 0.003
RankUp [18]	10.251 ± 0.072	0.599 ± 0.005	8.836 ± 0.047	0.676 ± 0.003	8.216 ± 0.022	0.703 ± 0.001
Ours	$\mathbf{9.177 \pm 0.061}$	$\mathbf{0.664 \pm 0.003}$	$\mathbf{8.539 \pm 0.065}$	$\mathbf{0.695 \pm 0.003}$	$\mathbf{8.166 \pm 0.071}$	$\mathbf{0.712 \pm 0.002}$

Table 3: Main results on STS-B. The performance is reported in the form of 'mean ± std' across six random runs. **Bold** means the best results and our method are shown in gray cells.

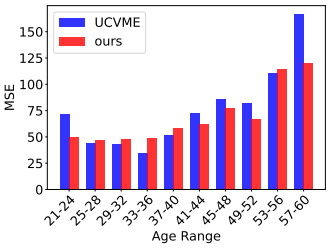
Method	$\gamma = 5\%$		$\gamma = 10\%$		$\gamma = 20\%$	
	MSE ↓	R^2 ↑	MSE ↓	R^2 ↑	MSE ↓	R^2 ↑
Fully-Supervised	0.986 ± 0.024	0.533 ± 0.012	0.986 ± 0.024	0.533 ± 0.012	0.986 ± 0.024	0.533 ± 0.012
Supervised	1.746 ± 0.016	0.173 ± 0.007	1.524 ± 0.010	0.278 ± 0.005	1.332 ± 0.012	0.369 ± 0.005
Mean Teacher [43]	1.751 ± 0.010	0.170 ± 0.005	1.607 ± 0.017	0.240 ± 0.007	1.409 ± 0.009	0.332 ± 0.004
Temporal Ensembling [26]	1.683 ± 0.007	0.203 ± 0.004	1.554 ± 0.009	0.264 ± 0.004	1.374 ± 0.015	0.349 ± 0.007
SSDKL [20]	1.606 ± 0.057	0.239 ± 0.027	1.407 ± 0.029	0.333 ± 0.014	$\mathbf{1.211 \pm 0.031}$	$\mathbf{0.426 \pm 0.015}$
TNNR [47]	1.746 ± 0.007	0.166 ± 0.004	1.534 ± 0.003	0.266 ± 0.001	1.333 ± 0.020	0.362 ± 0.010
SimRegMatch [21]	1.714 ± 0.032	0.188 ± 0.015	1.559 ± 0.024	0.261 ± 0.011	1.437 ± 0.015	0.319 ± 0.015
UCVME [10]	1.713 ± 0.009	0.188 ± 0.004	1.577 ± 0.016	0.253 ± 0.008	1.373 ± 0.020	0.349 ± 0.009
CLSS [9]	1.622 ± 0.046	0.231 ± 0.022	1.421 ± 0.030	0.327 ± 0.014	1.286 ± 0.032	0.390 ± 0.015
RankUp [18]	1.844 ± 0.063	0.126 ± 0.030	1.454 ± 0.016	0.311 ± 0.008	1.279 ± 0.018	0.394 ± 0.008
Ours	$\mathbf{1.540 \pm 0.006}$	$\mathbf{0.270 \pm 0.003}$	$\mathbf{1.403 \pm 0.013}$	$\mathbf{0.335 \pm 0.006}$	1.246 ± 0.015	0.409 ± 0.008

4.2 Main Results

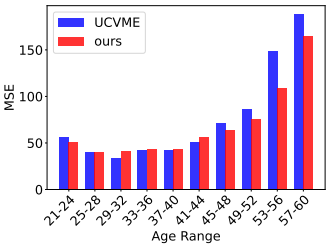
Results on UTKFace. Table 1 summarizes the results of our method and other comparison methods on UTKFace dataset with various ratios. It can be observed that: 1) All SSR methods perform better than the benchmark *Supervised*, which is the lower bound trained by labeled samples, demonstrating that incorporating unlabeled data is beneficial for the SSR problem and effectively reduces regression error. For example, under the ratio of 5%, our method reduces the MAE by up to 8.1% and increases R^2 by up to 15.2% compared to *Supervised*. 2) Compared to other SSR methods, our method consistently achieves the best or second best results across all ratios. In particular, when the labeled data is scarce ($\gamma = 5\%$), our method outperforms the second best RankUp [18] by 1.4% for MAE and 5.7% for R^2. 3) As the proportion of labeled data increases, the performance gains of our method are slightly lower than those of UCVME and RankUp. We attribute this to stronger supervision from labeled data, which likely improves the quality and stability of pseudo-labels. Such results further demonstrate that the well-calibrated uncertainty generated by our method plays an essential role in improving SSR performance, especially in scenarios where the labeled data is scarce.

Results on IMDB-WIKI. As shown in Table 2, our method consistently outperforms all comparison SSR methods significantly. For instance, under $\gamma = 5\%$, compared to the second best method Mean Teacher, our method reduces MAE by about 3.3% and increases R^2 by 2.6%, which achieves new state-of-the-art results. Note that the performance of our method with 20% labeled data is closer to *Fully-Supervised*, served as an upper bound in the SSR problem. These results further validate the strength of our method. Furthermore, compared with other uncertainty estimation SSR methods like UCVME, our method exhibits significant improvement, demonstrating the effectiveness of our predicted uncertainty by g_ϕ in our bi-level optimization framework.

Results on STS-B. The results are listed in Table 3. It can be observed that our method achieves the best or second-best performance across all ratios. On the one hand, if the labeled data is limited, such as $\gamma = 5\%$ or 10%, our method improves all these SSR methods on both MSE and R^2 with a large



(a) Predictions on unlabeled data



(b) Predictions on test data

Figure 3: Subgroup performance comparison between ours and the second-best (UCVME) on UTKFace with $\gamma = 5\%$.

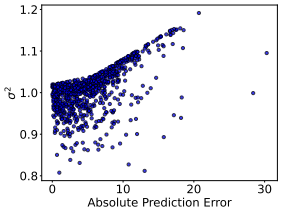


Figure 4: Correlation analysis between estimated uncertainty σ^2 and absolute prediction error on IMDB-WIKI with $\gamma = 10\%$ under age 40.

Table 4: Ablation study results on the IMDB-WIKI dataset with label ratios of 5% and 10%. **BL**, **UL**, and **BLO** refer to *Baseline*, Uncertainty-learner and Bi-level Optimization, respectively.

Components			$\gamma = 5\%$		$\gamma = 10\%$	
BL	UL	BLO	MAE ↓	R^2 ↑	MAE ↓	R^2 ↑
✓	×	×	9.512	0.651	8.864	0.683
✓	✓	×	9.914	0.630	9.562	0.651
✓	✓	✓	**9.177**	**0.664**	**8.539**	**0.695**

Table 5: Computational cost (average training time per iteration) and memory complexity on UTKFace dataset with $\gamma = 10\%$.

Method	Time (ms)	GPU (MB)
Baseline	83.0	5099
SimRegMatch[21]	548.1	7419
UCVME [10]	257.2	10057
RankUp[18]	108.6	7433
Ours	91.6	5116

margin. For example, when we have only 5% labeled data, MSE of our method is lower than the second-best method SSDKL by 4.1% and R^2 is higher by 13.0%. These results demonstrate that our method does not rely on too much labeled data and performs better under such extreme scenarios with scarce labeled samples. On the other hand, our method achieves comparable results with SSDKL when we have more labeled data, *i.e.*, $\gamma = 20\%$. This suggests that our method is not only effective in data-scarce scenarios but also highly adaptable to different levels of labeled data availability.

4.3 Discussion and Ablation Study

Performance analysis across subgroups. As aforementioned, our method consistently achieves sound performance across three different datasets with various ratios. To further validate the effectiveness of our method, in Fig. 3, we compare the MSE between the ground truth labels and pseudo-labels estimated by UCVME and our method across all ages on UTKFace. We can see that our method can globally produce more accurate pseudo-labels than UCVME across different subgroups, especially on the elder age ranges whose samples are significantly fewer than those of other age ranges, as shown in Fig. 1. These results further demonstrate that our method can consistently improve the accuracy across different subgroups.

How our algorithm adjusts uncertainty. To answer this question, we visualize the correlation between the uncertainty estimated by the proposed uncertainty-learner g_ϕ and the prediction error of the backbone regression model f_θ. Specifically, for all unlabeled training samples of IMDB-WIKI at age 40, we visualize their corresponding estimated uncertainties σ^2 and absolute prediction errors in Fig. 4. It can be observed that the estimated uncertainties increase as the prediction errors increase, indicating that the uncertainty-learner tends to assign larger uncertainty for the samples with larger prediction errors and actively reduces the contribution of these samples to the regression model training. Furthermore, the correlation between estimated uncertainties and absolute prediction errors is not a simple linearity, suggesting it is essential to use the uncertainty-learner to estimate the unstable uncertainties of different samples. Additional visualization results are provided in Appendix C.1.

Ablation study. We conduct an ablation study to analyze the contribution of each component of our method, with the results presented in Table 4. We begin with the *Baseline* (BL) method, which omits the uncertainty-learner (UL) and applies the loss $\mathcal{L}_l + \lambda\mathcal{L}_u$ with a fixed uncertainty $\sigma_j^2 = 1$ for all unlabeled samples. Compared to our method, the absence of uncertainty estimation leads

to poor performance, confirming its effectiveness in SSR. Next, we assess the roles of the UL and bi-level optimization (BLO) in our method. Interestingly, the model with UL alone (*Baseline* + UL) performs worse than the *Baseline*, likely due to joint training of the UL and the regression model, which leads to inaccurate uncertainty estimation and suppress learning from hard but correct samples, as discussed in Section 3.3. In contrast, incorporating the proposed bi-level optimization framework substantially improves performance, suggesting that it enables more accurate uncertainty estimation and, consequently, more effective training of the regression model. This complies with the conclusion of Theorem 1, which theoretically supports the benefit of bi-level optimization in improving uncertainty estimation.

Computational cost analysis. We evaluate the computational cost on a single NVIDIA GeForce RTX 4090 for fair comparison. As shown in Table 5, our method incurs only ~17MB additional GPU memory and less than 9ms extra training time per iteration compared to the *Baseline*, as it unrolls gradients only through the linear regression head to update a small number of parameters ϕ in the outer-loop optimization. Compared to recent SSR methods, our method achieves superior performance on all datasets with less time and resource consumption. Further analysis of the computational cost of other recent SSR methods is provided in Appendix C.2.

5 Conclusion

We propose an uncertainty-aware pseudo-labeling framework for SSR tasks, which addresses the challenge of heteroscedastic noise in pseudo-labels. Different from the existing methods, our approach dynamically calibrates pseudo-label uncertainty from a bi-level optimization perspective. Such learning paradigm ensures reliable uncertainty estimation that improves generalization of the regression model. Theoretical analysis via gradient alignment and empirical results on benchmark SSR tasks demonstrate the effectiveness of our method, significantly enhancing robustness and accuracy over existing approaches.

Limitations and Broader Impact. Despite the promising results achieved by our method in semi-supervised regression, particularly in enhancing the robustness of pseudo-labels through heteroscedastic uncertainty modeling, several limitations remain to be addressed. Notably, our approach does not explicitly consider potential systematic biases present in the labeled or unlabeled data, such as demographic imbalances or domain-specific disparities. If left unmitigated, these biases may be implicitly propagated or even amplified through the pseudo-labeling process. Addressing these issues will be one of the key focuses of our future work. Moreover, our current framework is built under the assumption that both labeled and unlabeled samples can be accessed simultaneously during training. Such a requirement may raise privacy concerns in real-world scenarios involving sensitive user data. Incorporating fairness-aware objectives and privacy-preserving mechanisms into future versions of our framework could further enhance its practical applicability and ethical reliability.

Acknowledgement

We would like to thank all anonymous reviewers for their constructive suggestions for improving this paper. This work was supported in part by the Major Key Project of PCL under Grant PCL2024A06, Tianyuan Fund for Mathematics of the National Natural Science Foundation of China under Grant 12426105, the China NSFC projects under contracts 62306233, 62476214 and 62372359, and the China Postdoctoral Science Foundation (2024M752550).

References

[1] 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.

[2] 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.

[3] C. M. Bishop and N. M. Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[4] N. Bodla, G. Hua, and R. Chellappa. Semi-supervised fusedgan for conditional image generation. In *Proceedings of the European Conference on Computer Vision*, pages 669–683, 2018.

[5] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics.

[6] X. Chen, Y. Yuan, G. Zeng, and J. Wang. Semi-supervised semantic segmentation with cross pseudo supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2613–2622, 2021.

[7] X. Cheng, D.-B. Wang, L. Feng, M.-L. Zhang, and B. An. Partial-label regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 7140–7147, 2023.

[8] 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.

[9] W. Dai, Y. Du, H. Bai, K.-T. Cheng, and X. Li. Semi-supervised contrastive learning for deep regression with ordinal rankings from spectral seriation. *Advances in Neural Information Processing Systems*, 36:57087–57098, 2023.

[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] R. Denton, S. Gross, and R. Fergus. Semi-supervised learning with context-conditional generative adversarial networks. *arXiv preprint arXiv:1611.06430*, 2016.

[12] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[13] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR, 2016.

[14] S. Haykin. *Neural networks: A comprehensive foundation*. Prentice hall PTR, 1994.

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[16] L. Hoyer, D. J. Tan, M. F. Naeem, L. Van Gool, and F. Tombari. SemiVL: Semi-supervised semantic segmentation with vision-language guidance. In *Proceedings of the European Conference on Computer Vision*, pages 257–275. Springer, 2024.

[17] X. Hu, Y. Niu, C. Miao, X.-S. Hua, and H. Zhang. On non-random missing labels in semi-supervised learning. In *International Conference on Learning Representations*, 2022.

[18] P.-Y. Huang, S.-W. Fu, and Y. Tsao. Rankup: Boosting semi-supervised regression with an auxiliary ranking classifier. *Advances in Neural Information Processing Systems*, 37:107444–107468, 2024.

[19] A. Immer, E. Palumbo, A. Marx, and J. Vogt. Effective bayesian heteroscedastic regression with deep neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

[20] N. Jean, S. M. Xie, and S. Ermon. Semi-supervised deep kernel learning: Regression with unlabeled data by minimizing predictive variance. *Advances in Neural Information Processing Systems*, 31, 2018.

[21] Y. Jo, H. Kahng, and S. B. Kim. Deep semi-supervised regression via pseudo-label filtering and calibration. *Applied Soft Computing*, 161:111670, 2024.

[22] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30, 2017.

[23] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.

[24] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[25] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

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

[27] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in Neural Information Processing Systems*, 30, 2017.

[28] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[29] D.-H. Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Proceedings of the Workshop on Challenges in Representation Learning, ICML*, volume 3, page 896. Atlanta, 2013.

[30] Z. Li, C. Deng, E. Yang, and D. Tao. Staged sketch-to-image synthesis via semi-supervised generative adversarial networks. *IEEE Transactions on Multimedia*, 23:2694–2705, 2020.

[31] Y. Lin, H. Yao, Z. Li, G. Zheng, and X. Li. Calibrating label distribution for class-imbalanced barely-supervised knee segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 109–118. Springer, 2022.

[32] H. Liu, K. Simonyan, and Y. Yang. Darts: Differentiable architecture search. In *International Conference on Learning Representations*, 2019.

[33] T. Mahmud, C.-H. Liu, B. Yaman, and D. Marculescu. Ssvod: Semi-supervised video object detection with sparse annotations. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6773–6782, 2024.

[34] S. L. Pintea, Y. Lin, J. Dijkstra, and J. C. van Gemert. A step towards understanding why classification helps regression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19972–19981, 2023.

[35] M. Ren, W. Zeng, B. Yang, and R. Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pages 4334–4343. PMLR, 2018.

[36] M. N. Rizve, K. Duarte, Y. S. Rawat, and M. Shah. In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning. In *International Conference on Learning Representations*, 2021.

[37] J. Rodemann, J. Goschenhofer, E. Dorigatti, T. Nagler, and T. Augustin. Approximately Bayes-optimal pseudo-label selection. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 1762–1773. PMLR, 2023.

[38] J. Rodemann, C. Jansen, G. Schollmeyer, and T. Augustin. In all likelihoods: Robust selection of pseudo-labeled data. In *Proceedings of the Thirteenth International Symposium on Imprecise Probability: Theories and Applications*, pages 412–425. PMLR, 2023.

[39] R. Rothe, R. Timofte, and L. Van Gool. Deep expectation of real and apparent age from a single image without facial landmarks. *International Journal of Computer Vision*, 126(2):144–157, 2018.

[40] M. Sajjadi, M. Javanmardi, and T. Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning. *Advances in Neural Information Processing Systems*, 29, 2016.

[41] 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.

[42] J. T. Springenberg. Unsupervised and semi-supervised learning with categorical generative adversarial networks. *arXiv preprint arXiv:1511.06390*, 2015.

[43] 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.

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

[45] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

[46] R. Wang, X. Jia, Q. Wang, Y. Wu, and D. Meng. Imbalanced semi-supervised learning with bias adaptive classifier. In *International Conference on Learning Representations*, 2023.

[47] S. J. Wetzel, R. G. Melko, and I. Tamblyn. Twin neural network regression is a semi-supervised regression algorithm. *Machine Learning: Science and Technology*, 3(4):045007, 2022.

[48] Y. Xia, F. Liu, D. Yang, J. Cai, L. Yu, Z. Zhu, D. Xu, A. Yuille, and H. Roth. 3d semi-supervised learning with uncertainty-aware multi-view co-training. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3646–3655, 2020.

[49] 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.

[50] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020.

[51] Z. Xing, Q. Dai, H. Hu, J. Chen, Z. Wu, and Y.-G. Jiang. Svformer: Semi-supervised video transformer for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18816–18826, 2023.

[52] X. Yang, Z. Song, I. King, and Z. Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8934–8954, 2023.

[53] Y. Yang, K. Zha, Y. Chen, H. Wang, and D. Katabi. Delving into deep imbalanced regression. In *International Conference on Machine Learning*, pages 11842–11851. PMLR, 2021.

[54] Y. Yang, D.-C. Zhan, Y.-F. Wu, Z.-B. Liu, H. Xiong, and Y. Jiang. Semi-supervised multi-modal clustering and classification with incomplete modalities. *IEEE Transactions on Knowledge and Data Engineering*, 33(2):682–695, 2019.

[55] H. Yao, X. Hu, and X. Li. Enhancing pseudo label quality for semi-supervised domain-generalized medical image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3099–3107, 2022.

[56] L. Yu, S. Wang, X. Li, C.-W. Fu, and P.-A. Heng. Uncertainty-aware self-ensembling model for semi-supervised 3d left atrium segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2019*, volume 11765 of *Lecture Notes in Computer Science*, pages 605–613. Springer, Cham, 2019.

[57] 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.

[58] Z. Zhang, Y. Song, and H. Qi. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5810–5818, 2017.

[59] Y. Zhou, X. He, L. Huang, L. Liu, F. Zhu, S. Cui, and L. Shao. Collaborative learning of semi-supervised segmentation and classification for medical images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2079–2088, 2019.

Supplementary Material

A Theoretical Analysis

A.1 Proof of Theorem 1

In this section, we first review the proposed bi-level learning framework to optimize the parameters of the regression model f_θ and the uncertainty-learner g_ϕ. The bi-level optimization framework can be formulated as follows:

$$\begin{aligned} \phi^* &= \arg\min_{\phi} \mathcal{L}^{outer}\left(\theta^*(\phi)\right) \\ \text{s.t.} \quad \theta^*(\phi) &= \arg\min_{\theta} \mathcal{L}^{inner}\left(\theta, \phi^*\right), \end{aligned} \tag{10}$$

where the specific inner and outer losses can be represented as

$$\begin{aligned} \mathcal{L}^{outer}\left(\theta(\phi)\right) &= \sum_{x_k^l \in \hat{\mathcal{B}}_l} \left(y_k - f_{\theta^*(\phi)}(x_k^l)\right)^2 \\ \mathcal{L}^{inner}\left(\theta, \phi\right) &= \mathcal{L}_l(\theta) + \lambda \mathcal{L}_u(\theta, \phi) \\ &= \sum_{x_i^l \in \mathcal{B}_l} \left(y_i - f_\theta(x_i^l)\right)^2 + \lambda \sum_{x_j^u \in \mathcal{B}_u} \left[\frac{1}{\exp(z_j)} \left(\hat{y}_j - f_\theta(x_j^u)\right)^2 + z_j\right], \end{aligned} \tag{11}$$

where $z_j = g_\phi(x_j^u)$ denotes the uncertainty estimated by the uncertainty-learner g_ϕ. To optimize this bi-level optimization framework, we use a nested gradient-optimization-based method to approximately update the regression model parameter θ and the uncertainty-learner parameter ϕ. Specifically, in the inner loop, the updating formulation of θ at iteration k can be expressed as:

$$\theta^{t+1}(\phi^t) = \theta^t - \alpha \cdot \nabla_\theta \mathcal{L}^{inner}(\theta^t, \phi^t), \tag{12}$$

where α is the learning rate of the inner loss.

$$\phi^{t+1} = \phi^t - \beta \cdot \nabla_\phi \mathcal{L}^{outer}(\theta^{t+1}(\phi^t)), \tag{13}$$

where β is the learning rate of the outer loss.

For simplicity of notation, we denote $\nabla_\theta \mathcal{L}^{inner}(\theta, \phi)$ and $\nabla_\theta \mathcal{L}^{outer}(\theta)$ be the gradients of the inner and outer loss w.r.t. θ, respectively. Then we provide the proof of Theorem 1 in the following.

$$\min_{\phi} - \left\langle \nabla_\theta \mathcal{L}^{inner}(\theta, \phi), \nabla_\theta \mathcal{L}^{outer}(\theta) \right\rangle. \tag{14}$$

Proof. For the outer optimization problem, we expand this loss function by the first-order Taylor expansion around θ^t as follows:

$$\mathcal{L}^{outer}(\theta^{t+1}(\phi^t)) \approx \mathcal{L}^{outer}(\theta^t) + \left(\theta^{t+1}(\phi^t) - \theta^t\right) \nabla_\theta \mathcal{L}^{outer}(\theta^t). \tag{15}$$

Substitute eq. (12) into eq. (15), we obtain

$$\mathcal{L}^{outer}(\theta^{t+1}(\phi^t)) = \mathcal{L}^{outer}(\theta^t) + \left(-\alpha \cdot \nabla_\theta \mathcal{L}^{inner}(\theta^t, \phi^t)\right) \nabla_\theta \mathcal{L}^{outer}(\theta^t). \tag{16}$$

Note that the first term on the right side of eq. (16) is irrelevant to the optimization of ϕ^t, which can be omitted. Thus, minimizing the outer loop of eq. (6) is equivalent to minimizing the second term in eq. (16). That is,

$$\min_{\phi^t} \mathcal{L}^{outer}\left(\theta^{t+1}(\phi^t)\right) \Leftrightarrow \min_{\phi^t} - \left\langle \nabla_\theta \mathcal{L}^{inner}(\theta^t, \phi^t), \nabla_\theta \mathcal{L}^{outer}(\theta^t) \right\rangle. \tag{17}$$

We thus finish the proof. □

A.2 Connection to Bayesian Decision Theory

We make an effort to connect our method to Bayesian decision theory [37, 38] to further interpret our bi-level framework. To make the theoretical analysis more tractable, we restate our bi-level learning formulation with a slight simplification. Specifically, we let the predictive variance $\sigma_j^2 = g_\phi(x_j)$, rather than using its logarithm as in the main text. This modification only involves a log-transform and does not alter the essential behavior of our model. The inner-loop optimization becomes:

$$\theta^*(\phi) = \arg\min_\theta \sum_{i=1}^{n} (y_i - f_\theta(x_i))^2 + \lambda \sum_{j=1}^{m} [\frac{1}{g_\phi(x_j)} (\hat{y}_j - f_\theta(x_j))^2 + \log g_\phi(x_j)], \tag{18}$$

and the outer-loop optimization is:

$$\phi^* = \arg\min_\phi \sum_{(x_k, y_k) \in \hat{\mathcal{B}}_l} (y_k - f_{\theta^*(\phi)}(x_k))^2, \tag{19}$$

which seeks optimal uncertainty weights $g_\phi(x_j)$ to guide the model $f_{\theta^*(\phi)}$ toward better generalization.

To link our bi-level learning framework to a Bayesian decision problem, we formulate the following decision problem. Concretely, the uncertainty estimator $g_\phi : \mathcal{X} \to \mathbb{R}^+$ defines a randomized (mixed) decision rule by assigning weights to unlabeled samples $x_j \in \mathcal{D}_u$. The model parameter θ is treated as the latent state, and we define the utility function as:

$$u(g_\phi, \theta) = \log P(\mathcal{D}_l \cup \mathcal{D}_u | \theta) = \log \prod_{x_i \in \mathcal{D}_l} \mathcal{N}(y_i | f_\theta(x_i), \sigma^2) \prod_{x_j \in \mathcal{D}_u} \mathcal{N}(\hat{y}_j | f_\theta(x_j), g_\phi(x_j)). \tag{20}$$

where $\hat{y}_j$ denotes the pseudo-label for x_j. Under this formulation, the Bayes-optimal uncertainty function g_ϕ^* is the one that maximizes the expected utility w.r.t the posterior distribution over θ, i.e., $\phi^* = \arg\max_\phi \mathbb{E}_{\theta|\mathcal{D}_l}[u(g_\phi, \theta)]$.

We now show an intuition connection between our bi-level optimization and this Bayesian formulation. Specifically, the inner-loop optimization is equivalent to minimizing the negative log-likelihood of the data under the generative model defined earlier. Therefore, $\theta^*(\phi)$ can be interpreted as a MAP estimate of model parameters θ, conditioned on the fixed uncertainty function g_ϕ. The outer-loop optimization aims to find the optimal uncertainty weights that lead to a model $f_{\theta^*(\phi)}$ with best generalization performance. While this is not a direct maximization of expected utility $\mathbb{E}_\theta[u(g_\phi, \theta)]$, we argue the following: (i) If the posterior $p(\theta|\mathcal{D}_l)$ is sharply peaked (posterior concentration), and (ii) if the outer-loop loss over $\hat{\mathcal{B}}_l$ is a reliable proxy for risk under $p(x, y)$, then minimizing the outer-loop loss approximately maximizes the expected utility, and thus ϕ^* approximates the Bayes-optimal mixed-action policy.

A.3 Pseudo-Label Smoothness Property

Our framework implicitly enforces a smoothness property on the generated pseudo-labels, as supported by both theoretical analysis and empirical evidence:

- **Theoretical smoothness guarantee.** The Lipschitz continuity of both the regression model f_θ and the uncertainty estimator g_ϕ ensures that nearby unlabeled inputs yield similar pseudo-labels and uncertainty estimates. Specifically, for any two unlabeled samples x_1^u and x_2^u, we have: $|\hat{y}_1 - \hat{y}_2| \leq L_f |x_1^u - x_2^u|$ and $|\sigma_1^2 - \sigma_2^2| \leq L_g L_f |x_1^u - x_2^u|$, where L_f and L_g are the Lipschitz constants of f_θ and g_ϕ, respectively. These constants can be effectively controlled by architectural design. For example, we can apply spectral normalization to f_θ to limit its Lipschitz constant, and we design g_ϕ as a shallow MLP with ReLU activations and small-initialized weights to promote smoothness.
- **Empirical evidence.** To further validate the smoothness of pseudo-labels in practice, we conducted a variance analysis across age groups on the IMDB-WIKI dataset. Specifically, we computed the standard deviation of predicted pseudo-labels among samples that share the same ground-truth age and compared the results with UCVME[10], which explicitly enforces local consistency via consistency loss. As reported in the table 6, our method consistently yields lower or comparable standard deviations than UCVME in most age groups. This suggests that our pseudo-labels are more stable and consistent across similar inputs, thereby indirectly capturing the desired smoothness prior.

Table 6: Performance comparison across age groups.

Method	Age=10	20	30	40	50	60	70	80	90
UCVME [10]	10.623	6.522	5.009	5.878	7.772	10.666	12.936	13.085	5.017
Ours	3.290	6.389	6.157	6.971	8.558	10.231	10.301	6.364	4.011

B Detailed Experimental Setup

B.1 Dataset Details

In this work, we experiment with three benchmark datasets: UTKFace [58], IMDB-WIKI [39], and STS-B [5, 45], which are detailed as follows:

UTKFace[58]. The dataset is an image age estimation dataset, where the objective is to predict an individual's age based on a facial image. The dataset consists of 23,705 face images with the labels ranging from 1 to 116 years old. Following the protocols used in UCVME [10], this paper considers only samples aged 21 to 60. The resulting modified dataset includes 10,518 training samples, 3,287 testing samples, and 2,629 validation samples.

IMDB-WIKI [39]. The dataset is a larger-scale age estimation dataset which collected over 523K facial images and their corresponding age. Following [53], we utilize curated datasets consisting of 191.5K images for training and 11.0K images for validation and testing. The ages range from 0 to 186 years old and the number of images per age varies from 1 to 7149.

STS-B [5, 45]. Semantic Textual Similarity Benchmark (STS-B) dataset, which consists of 7.2K sentence pairs collected from real worlds, such as news, image and video captions, and natural language inference data. The target of each pair is a continuous similarity score ranged from 0 to 5. Following DIR [53], we construct the training set containing 5.2K pairs, and both the balanced validation set and the test set containing 1K pairs each.

B.2 Evaluation Metrics

To assess the performance of comparison algorithms, we adopt Mean Absolute Error (MAE $\downarrow$) and the coefficient of determination ($R^2 \uparrow$) for image datasets following UCVME [10], while employing Mean Squared Error (MSE $\downarrow$) and R^2 metrics for text datasets following DIR [53].

Mean Absolute Error (MAE $\downarrow$). MAE measures the average absolute difference between predictions and ground truths, offering a simple and robust metric for evaluating prediction accuracy. *i.e.*,

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|f_{\theta^*}(x_i) - y_i\right|, \tag{21}$$

where $f_{\theta^*}(x_i)$ denotes the prediction for the sample x_i, and y_i is the corresponding ground truth.

Coefficient of Determination ($R^2 \uparrow$). R^2 reflects the goodness of fit of the model to the data. *i.e.*,

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(f_{\theta^*}(x_i) - y_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}, \tag{22}$$

where $\overline{y}$ is the mean of y_i.

Mean Squared Error (MSE $\downarrow$). MSE measures the average squared difference between predictions and ground truths, emphasizing larger errors due to the squaring operation. *i.e.*,

$$\text{MSE} = \frac{1}{n}\sum_{i}^{n}(f_{\theta^*}(x_i) - y_i)^2. \tag{23}$$

B.3 Training Details

UTKFace and IMDB-WIKI. Following the training protocol in UCVME [10], we utilize ResNet-50 [15] pretrained on the ImageNet dataset as the backbone regression model. The model is trained by Adam optimizer [24] for 30 epochs, with a learning rate of 10^{-4} for the feature extractor and 10^{-3} for the regression head. Additionally, we conduct random cropping and horizontal flipping as

weak augmentation, and RandAugment [8] as strong augmentation in the data augmentation process of SSL. As for g_ϕ, it is optimized by Adam optimizer with a learning rate of 10^{-4}.

STS-B. Following [45, 53], we adopt a BiLSTM architecture with GloVe word embeddings as the backbone regression model, which is trained by Adam optimizer for 200 epochs, with a learning rate of 10^{-4} for the feature extractor and 10^{-3} for the regression head. For data augmentation, no augmentation is applied to the labeled dataset, while synonym replacement and insertion operations are used as strong augmentation for the unlabeled dataset.

B.4 Comparison Methods

To comprehensively evaluate the proposed method, we compare it with some state-of-the-art semi-supervised regression methods, including: Consistency-based methods: Mean Teacher [43], Temporal Ensembling [26], TNNR [47], UCVME [10], CLSS [10], RankUp [18]; Uncertainty-based methods: SSDKL [20], SimRegMatch [21].

In this section, we provide a brief introduction to comparison algorithms.

- **Mean Teacher** [43] averages model weights to form a target-generating teacher model via exponential moving average (EMA), and computes the consistency loss between the teacher's predictions and the student's outputs.
- **Temporal Ensembling** [26] maintains an exponential moving average of label predictions on each training example, and penalizes predictions that are inconsistent with this target.
- **TNNR** [47] is trained to predict differences between the labels of the input pair and follow the principle that the loop of predicted differences should sum to zero.
- **UCVME** [10] improves training by generating high-quality pseudo-labels and uncertainty estimates for heteroscedastic regression.
- **CLSS** [9] uses the recovered ordinal relationship for contrastive learning on unlabeled samples to allow more data to be used for feature representation learning.
- **RankUp** [18] converts the original regression task into a ranking problem and training it concurrently with the original regression objective. It introduces two components: the Auxiliary Ranking Classifier (ARC) and Regression Distribution Alignment (RDA).
- **SSDKL** [20] is a non-parametric kernel learning methods based on minimizing predictive variance in the posterior regularization framework. It combines the hierarchical representation learning of neural networks with the probabilistic modeling capabilities of Gaussian processes.
- **SimRegMatch** [21] is a semi-supervised regression framework with two primary modules: uncertainty-based filtering and similarity-based pseudo-label calibration. It filters reliable pseudo-labels using uncertainty and refines them by incorporating labeled data through similarity-based weighting.

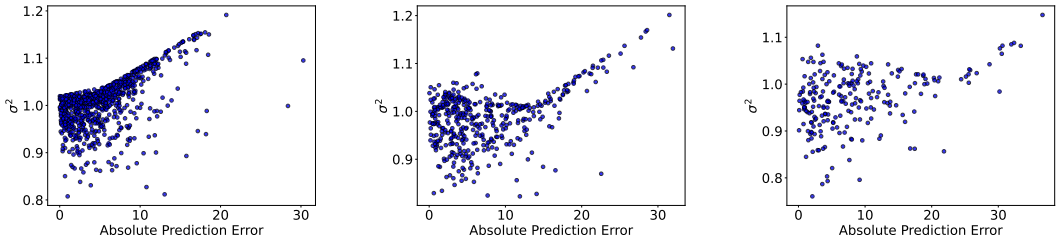
C Additional Experiment Results

C.1 More Visualization Results

To further analyze the correlation between estimated uncertainty σ^2 and absolute prediction error, we present additional age-wise visualizations on the IMDB-WIKI dataset. Consistent with observations in Fig. 5, the uncertainty σ^2 increases alongside the absolute prediction error across age groups, highlighting the role and effectiveness of the uncertainty-learner.

C.2 Computational Resources

Compared to other SSR methods, our proposed algorithm achieves superior performance while maintaining significantly lower computational and memory costs. Among the algorithms compared in table 5: SimRegMatch [21] estimates pseudo-label uncertainty through Monte Carlo sampling, which requires multiple forward passes, and stores feature vectors of samples to compute the similarity matrix. UCVME [10] relies on a multi-model architecture and aggregates predictions from repeated forward passes to compute uncertainty-based consistency loss. RankUp [18] not only



(a) age=40 (b) age=50 (c) age=60

Figure 5: Correlation analysis between estimated uncertainty σ^2 and absolute prediction error on IMDB-WIKI with $\gamma = 10\%$ under different age.

outputs regression predictions, but also generates ranking scores for each sample, necessitating the computation of all pairwise ranking losses. Furthermore, its RDA module involves frequent lookups from a pseudo-label table during training, which adds additional time overhead. Our method does not require multiple forward passes or the storage of additional intermediate variables during computation. Meanwhile, the proposed uncertainty-learner is a lightweight network; in the optimization process, we assume that the parameters of the uncertainty-learner are only related to the regression head, which significantly reduces computational time and resource consumption.

C.3 Additional Experimental Comparisons

To further assess the generality and robustness of our method, we additionally include methods from weakly supervised learning (PIM [7]), semi-supervised classification (UPS [36] and CADR [17]).

We provide a brief introduction to these methods below.

- **PIM** [7] computes its loss as a weighted sum of the predictive losses over candidate labels, where labels with smaller losses are assigned higher weights through a softmax-based weighting function.
- **UPS** [36] uses Monte Carlo Dropout to estimate uncertainty via the variance of multiple forward passes, and applies a threshold to identify low-confidence predictions as negative labels, thereby enabling negative learning.
- **CADR** [17] leverages the prior distribution of labeled samples to perform inverse probability weighting (CAP module) and adjusting per-class selection thresholds for unlabeled samples (CAI module).

Adapting existing methods for SSR. Several of the compared methods cannot be directly applied to regression tasks, so we propose reasonable adaptations to enable a fair comparison under the SSR setting. Specifically, (1) PIM is originally designed for partial-label regression, where each unlabeled instance is associated with a candidate label set. Since such candidate sets are absent in semi-supervised regression, we construct pseudo-label candidates via Monte Carlo Dropout and compute the PIM loss using a softmax-based weighting over these candidates. (2) UPS relies on defining negative targets in a discrete label space, which is not directly applicable to continuous regression outputs. To adapt UPS fairly, we retain only the uncertainty-based pseudo-label estimation via Monte Carlo Dropout and remove the negative labeling mechanism, yielding the adapted variant UPS_Reg. (3) CADR builds on class-specific confidence modeling, which presupposes discrete label categories. To make it applicable to regression, we discretize the continuous target into age bins and treat each bin as a surrogate class for confidence estimation.

Results Analysis. As shown in table 7, our proposed method significantly outperforms all comparison methods across multiple settings. Under the ratio of 20%, our method outperforms the second best method PIM by 3.3% for MAE and 1.1% for R^2.

Comparison with weighting-based methods. Our uncertainty-aware method and PIM both share the common idea of assigning different loss contributions to samples. However, there is a fundamental difference in how the weights are generated. PIM assigns weights solely based on the prediction

Table 7: Experimental comparison with more methods on the IMDB-WIKI dataset. **Bold** means the best results and our method are shown in gray cells.

Method	$\gamma = 5\%$		$\gamma = 10\%$		$\gamma = 20\%$	
	MAE ↓	R^2 ↑	MAE ↓	R^2 ↑	MAE ↓	R^2 ↑
PIM [7]	9.345 ± 0.100	0.659 ± 0.005	8.691 ± 0.021	0.691 ± 0.001	8.441 ± 0.014	0.704 ± 0.001
UPS_Reg [36]	9.430 ± 0.087	0.647 ± 0.006	8.845 ± 0.055	0.676 ± 0.003	8.415 ± 0.002	0.699 ± 0.001
CADR [17]	10.592 ± 0.091	0.562 ± 0.009	9.531 ± 0.064	0.622 ± 0.007	8.536 ± 0.056	0.668 ± 0.008
Ours	**9.177 ± 0.061**	**0.664 ± 0.003**	**8.539 ± 0.065**	**0.695 ± 0.003**	**8.166 ± 0.071**	**0.712 ± 0.002**

error of pseudo-labels within each batch, making it prone to reinforcing inaccurate pseudo-labels and amplifying noise. In contrast, our method infers per-sample uncertainty and adaptively adjusts each sample's contribution in a continuous manner, resulting in more reliable weighting.

Comparison with classification-derived SSL methods. The inferior performance of UPS and CADR highlights the challenge of directly adapting classification-based methods to regression tasks. Morevoer, compared with UPS_Reg, while both methods incorporate uncertainty, our method uses uncertainty-aware soft weighting within a bi-level optimization framework, yielding more stable and accurate uncertainty estimation.

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 claims reflect the paper's contributions, which is proposing an uncertainty-aware pseudo-labeling framework that dynamically adjusts pseudo-label influence from a bi-level optimization perspective. Our empirical results and theoretical insights demonstrate superior robustness and performance compared to existing methods.

   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 5.

   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?

Answer: [Yes]

Justification: We have provide full set of assumptions and a complete (and correct) proof in Appendix A.1.

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 and provided some experiments details in Appendix B.

   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:All the datasets are public as illustrated in Appendix B.1. We will release the full code and commands once the paper gets accepted.

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: We claim the experiments setting and detainls in Appendix B.

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 conducted six runs using fixed random seeds (0, 1, 2, 3, 4, and 5), and we report both the mean and standard deviation of each metric.

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 Section 4.3.

   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 verified that all requirements of the ethical guidelines have been met.

   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: [Yes]

    Justification: We have discussed the broader impacts of our proposed method in Section 5.

    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 does not release any high-risk data or models.

    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 code, data, and models used in this work to ensure transparency and proper acknowledgment.

    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.
    - 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 code used in this paper, which is well documented. The documentation includes detailed descriptions of the code's purpose, functionality, usage examples, and license details. The documentation will be publicly available.

    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: This paper does not involve human subjects or crowdsourcing experiments.

    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 study does not involve 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.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The paper did not describe the usage of LLMs, as they were only used for writing, editing, or formatting purposes.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.