HISTOGRAM-GUIDED SOURCE-FREE DOMAIN ADAPTIVE REGRESSION

Anonymous authors
Paper under double-blind review

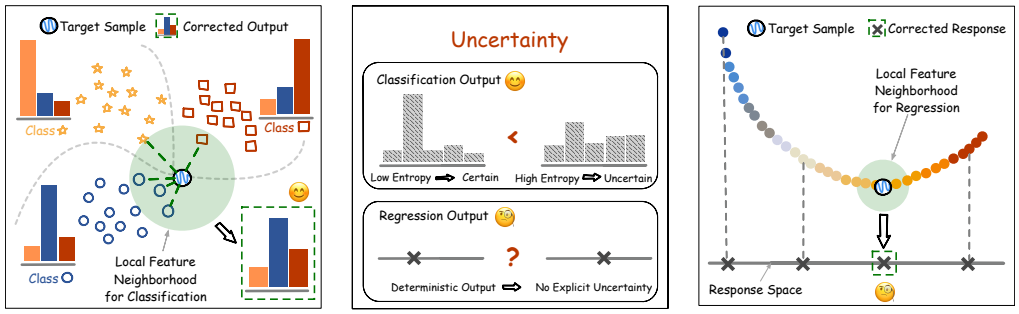
ABSTRACT

Source-Free Domain Adaptation (SFDA) enables model adaptation under distribution shifts without access to source data, making it an appealing solution for privacy-sensitive applications. Despite being a fundamental problem in machine learning, regression remains largely underexplored in SFDA, where most existing work has focused predominantly on classification tasks. To bridge this gap, we propose a novel algorithm that leverages sample-wise, histogram-informed supervisory signals to refine pseudo-labels under an uncertainty-aware paradigm. This design simultaneously achieves pseudo-label refinement and uncertainty modeling, two key components that are critical for effective adaptation in classification but remain largely absent in regression. We further theoretically show that the resulting histograms exhibit robustness to potential perturbations, supporting reliable SFDA for regression. Empirical results across multiple benchmarks confirm the effectiveness of our method and reveal that histogram-guided learning promotes more compact and structured feature representations, mitigating the inherent challenges of adapting regression models under distribution shift.

1 INTRODUCTION

With growing needs for model adaptivity and the concerns about data privacy, Source-Free Domain Adaptation (SFDA) has emerged as a practical solution, allowing models to adapt to a target domain without requiring access to source data. A key focus of recent research in SFDA for classification tasks lies in constructing informative supervision signals by exploiting the geometric structure of the feature space and the intrinsic uncertainty in the output space. Specifically, the clustered structure of the feature space, induced by the discrete and non-ordinal nature of class labels, offers topological cues for refining pseudo-labels, as local neighborhoods often contain diverse supervision signals that help correct ambiguous predictions, particularly those near decision boundaries (Liang et al., 2020; Yang et al., 2022b; 2021) (Figure 1a). Meanwhile, the softmax distribution in the output space naturally reveals sample-level uncertainty (Figure 1b), and facilitates confidence-aware self-training strategies, e.g., guiding feature alignment and further pseudo-label correction in a curriculum-based manner (Zhang et al., 2022b; Mitsuzumi et al., 2024).

In contrast, regression tasks pose fundamentally different challenges. Notably, the local manifold learned by regression typically maps to smoothly varying, continuous, and inherently ordered output values. As a result, nearby points in the feature space tend to share similar predictions, offering limited supervision signals for pseudo-label refinement (Figure 1c). In addition, deep regression models are inherently deterministic: the regression head outputs only a single scalar prediction for each input, providing no estimate of prediction variance or confidence. This absence of explicit uncertainty makes it difficult to assess prediction reliability or perform label correction (Figure 1b). This issue is further exacerbated under distribution shift, where the model's predictions become less reliable, compounding the difficulty of adaptation. Moreover, recent studies (Zhang et al., 2024; 2025) show that typical losses (e.g. MSE, L_1) for deep regression models tend to induce a lower marginal entropy of feature representations Z while minimizing the conditional entropy $\mathcal{H}(\mathrm{Z}|\mathrm{Y})$ given the target variable Y. Such entropy-reducing properties limit feature generalizability and hinder robust representation learning under inaccurate labels, making it even more challenging to adapt to the target domain in the absence of source data and target labels. Consequently, Source-Free Domain Adaptive Regression (SFDAR) remains largely underexplored and demands new approaches tailored to its unique challenges.



(a) Classification: Clustered Feature Space (b) Uncertainty in Outputs (c) Regression: Smooth Feature Space

Figure 1: *This figure highlights key differences between classification and regression tasks under SFDA. (a) shows how the clustered feature space in classification supports neighborhood-based label correction, aided by distributional output information (distinct color–shape markers indicate different classes). In contrast, (c) illustrates that the smooth feature space in regression restricts the effectiveness of such refinement strategies (the gradually changing colors represent continuous target values). (b) further demonstrates that classification outputs naturally encode uncertainty (e.g., entropy of class probabilities), while regression outputs are typically deterministic and lack explicit distributional information (e.g., entropy or confidence estimates).*

To this end, we propose Mutual Enhancement of Regression-Classification Integration (MERCI), a novel framework that introduces histogram-informed supervisory signals to support SFDAR. MERCI adopts a bifurcated architecture: alongside the primary regression head, it introduces a histogram head to model the sample-wise conditional distribution of the target label. This module can further guide and regularize regression predictions through cross-head interactions, as illustrated in Figure 2. More specifically, to enable effective histogram head training, an initial uncertainty-aware label set is constructed via multiple stochastic forward passes (e.g., dropout) through the regression model. In parallel, a unimodal prior is imposed to reflect the inherent continuity and smoothness of regression outputs. The resulting discrete histogram distributions thus enable the construction of informative, continuous supervision signals via their expected values, which in turn enhance the training of the target regression model. The benefits of incorporating histogram signals for SFDAR are twofold. First, they introduce uncertainty-awareness into regression and provide rich distributional cues for pseudo-label refinement. Second, the distributional losses imposed by the histogram head are more effective at tightening the learned feature representations (Zhang et al., 2025), as shown in Figures 4 and E.6, thereby resolving the aforementioned fundamental challenges in SFDAR and improving adaptation performance. In addition, we theoretically show that the resulting histograms exhibit certain robustness to potential perturbations, which further supports the reliability and the effectiveness of MERCI for SFDAR.

Our main contributions can be summarized as follows:

1) We propose MERCI, a novel framework for SFDAR that integrates histogram-based modeling into regression. MERCI refines the supervision signal to guide the adaptation process by capturing the label uncertainty and further strengthens generalization by encouraging feature compression.
2) We design a structured histogram learning module that supports regression adaptation without labeled target data, and provide theoretical analysis to support its robustness and effectiveness.
3) We conduct extensive experiments across diverse regression tasks, along with detailed ablation studies, to demonstrate the effectiveness and reliability of MERCI.

2 Related Work

Unsupervised Domain Adaptive Regression. Traditional domain adaptation methods for regression typically assume access to labeled source data and unlabeled target data, aiming to improve model performance in the target domain. Mainstream approaches reduce domain discrepancy through

adversarial training or explicit feature alignment (Ganin et al., 2016; Chen et al., 2021). For example, RSD (Chen et al., 2021) highlights the sensitivity of regression models to feature scales and minimizes subspace distances accordingly. To improve stability, DARE-GRAM (Nejjar et al., 2023) aligns the inverse Gram matrices of source and target features. Other approaches leverages the training dynamics of distribution-informed neural networks with MMD in NTK-induced RKHS (Wu et al., 2022), or perform uncertainty-aware feature alignment (Nejjar et al., 2024). However, all these methods require access to raw source data during adaptation. To address privacy and security concerns, recent efforts such as TASFAR (He et al., 2024) and SSA (Adachi et al., 2024) attempt to eliminate the use of raw source data, but still depend on source-trained modules or statistics, and thus do not fully adhere to source-free constraint. Related studies in other modalities, including energy-based test-time adaptation for depth completion Chung et al. (2025); Wang et al. (2024), non-stationarity-aware test-time adaptation for time-series forecasting Kim et al. (2025), and fully test-time adaptation for tabular prediction Zhou et al. (2025), demonstrate the increasing interest in adapting models under distribution shift without source data. Although these methods offer useful insights, their architectures and problem formulations are tailored to specific tasks and cannot be directly applied to the SFDAR setting considered here. In this work, we consider a challenging Source-Free Domain Adaptive Regression (SFDAR) setting, where only a pre-trained source model is available for adaptation. Further details of the SFDAR problem, including its real-world applications and general types of distribution shift, are provided in Appendix D.2.

Classification in Source-Free Domain Adaptation (SFDA). SFDA for classification tasks has been extensively studied and can be broadly categorized into two branches: pseudo-label correction and contrastive representation-based self-training. Among them, SHOT (Liang et al., 2020) proposes a source hypothesis transfer strategy based on entropy minimization and deep clustering. Subsequent works extend SHOT by improving feature structure through neighborhood consistency (Yang et al., 2022b; 2021) or contrastive learning (Zhang et al., 2022b; Mitsuzumi et al., 2024). These methods all rely on the inherently clustered structure of the classification feature space and the uncertainty encoded in softmax outputs, which together facilitate confidence-aware pseudo-labeling and self-training. However, such assumptions break down in regression tasks, where the feature space lacks clear clustering and the output is continuous and deterministic.

Regression as Classification. Recently, classification-style losses, such as cross-entropy, have been introduced into regression-related tasks, and have demonstrated promising performance (Cao et al., 2017; Sun et al., 2025; Imani & White, 2018). Zhang et al. (2023) and Imani et al. (2024) analyze the benefits of incorporating such distributional losses into regression tasks from the perspectives of feature representation and gradient stability, respectively, further validating the importance of learning a distribution over target labels, rather than predicting a single point estimate. However, these methods and theoretical findings are limited to fully supervised learning scenarios with accurate response values. In contract, the challenge in SFDAR lies in effectively constructing meaningful classification proxies and generating supervision signals without source data. In our study, we propose leveraging histograms as a bridge between classification and regression to address this challenge.

3 Problem Setup

We consider a regression problem with input space $\mathcal{X} \subseteq \mathbb{R}^d$ and target space $\mathcal{Y} \subseteq \mathbb{R}$, where d denotes the input dimension. In the SDFA setting, we assume that the source and target domain distributions, $P^{\mathrm{S}}_{\mathrm{xy}}$ and $P^{\mathrm{T}}_{\mathrm{xy}}$, are unknown and potentially different. These distributions are defined over the joint space $\mathcal{X} \times \mathcal{Y}$, and can be factorized into marginal and conditional components as follows: $P^{\mathrm{S}}_{\mathrm{xy}} = P^{\mathrm{S}}_{\mathrm{x}} P^{\mathrm{S}}_{\mathrm{y|x}}$ and $P^{\mathrm{T}}_{\mathrm{xy}} = P^{\mathrm{T}}_{\mathrm{x}} P^{\mathrm{T}}_{\mathrm{y|x}}$, respectively. In SFDA, we are given a *source regression model* $h_{\mathrm{S}} : \mathcal{X} \to \mathcal{Y}$ pretrained on a *labeled source dataset* $\mathcal{D}_{\mathrm{S}} \triangleq \{\mathrm{x}^{\mathrm{S}}_i, \mathrm{y}^{\mathrm{S}}_i\}_{i=1}^{N_{\mathrm{S}}}$ sampled from $P^{\mathrm{S}}_{\mathrm{xy}}$, along with an *unlabeled target dataset* $\mathcal{D}_{\mathrm{T}} \triangleq \{\mathrm{x}^{\mathrm{T}}_i\}_{i=1}^{N_{\mathrm{T}}}$ drawn from $P^{\mathrm{T}}_{\mathrm{x}}$. Given access to h_{S} and $\mathcal{D}_{\mathrm{T}}$, our objective is to learn a *target regression model* $h_{\mathrm{T}} : \mathcal{X} \to \mathcal{Y}$ that performs well on the target domain by adapting h_{S} on $\mathcal{D}_{\mathrm{T}}$, without access to the original source data $\mathcal{D}_{\mathrm{S}}$.

We denote the source and target models as $h_{\mathrm{S}} = f_{\mathrm{reg,S}} \circ g_{\mathrm{S}}$ and $h_{\mathrm{T}} = f_{\mathrm{reg,T}} \circ g_{\mathrm{T}}$, where g_{S} and g_{T} are feature extractors, $f_{\mathrm{reg,S}}$ and $f_{\mathrm{reg,T}}$ represent the corresponding regression heads, and $\circ$ denotes the functional composition operator. In practice, the source and target models h_{S} and h_{T} typically share the same network architecture, with the target model initialized from the source model. We therefore use the generic notation $h = f_{\mathrm{reg}} \circ g$ to refer to either model unless a distinction is necessary.

Notations. For any $v \in \mathbb{R}^q$, we let $\|v\| \triangleq \sqrt{\sum_{j=1}^{q} v_j^2}$ denote its ℓ_2 norm. For any positive integer K, we denote $[K] \triangleq \{1, \ldots, K\}$, and define the $(K-1)$-dimensional probability simplex as: $\Delta^{K-1} \triangleq \{\mathrm{s} \in \mathbb{R}^K : s_j \geq 0, \sum_{k=1}^{K} s_j = 1\}$. For any $p \in \Delta^{K-1}$, we also use $p[j]$ to denote its jth component for $j \in [K]$. We use capital letters (e.g., X^{T}, Y^{T}) to denote random variables, and the corresponding lowercase letters (e.g., x, y, x^{T}, y^{T}) to denote their realizations. For any discrete set $\mathcal{S}$, we use $|\mathcal{S}|$ to denote its cardinality.

4 Methodology

Our framework, MERCI, employs a dual-head architecture comprising a regression head (f_{reg}) and a classification tasked-based histogram head (f_{hist}), both built upon a shared feature extractor (g). Importantly, the histogram head in our framework goes beyond conventional classification tasks: it approximates a discretized version of the conditional density $\mathbb{P}(\mathrm{Y}^{\mathrm{T}}|\mathrm{X}^{\mathrm{T}} = \mathrm{x})$ for the target label Y^{T} given input $\mathrm{X}^{\mathrm{T}} = \mathrm{x}$. By leveraging the *ordered* and *continuous* nature of regression outputs, this head allows for better uncertainty characterization and facilitates more robust adaptation in the absence of source data and target label. Moreover, the loss component associated with the histogram head also implicitly encourages the learned features to compress irrelevant label information, which in turn improves generalization performance across domains (Zhang et al., 2025).

An overview of the framework is shown in Figure 2. It operates through *bi-directional knowledge transfer* between the regression and histogram heads, consisting of two key components: (1) For each target input x, we perform multiple stochastic forward passes through the regression head (e.g., via dropout) to generate an ensemble of predictions. These predictions are used to construct a partial label set and build a unimodal prior distribution, enabling the learning of a histogram that approximates the conditional target label distribution (Section 4.1); (2) The predictive distribution from the histogram head is then used to generate a continuous pseudo-label ($\bar{\mathrm{y}}$) combining the uncertainty estimation, which in turn refines the regression head (Section 4.2).

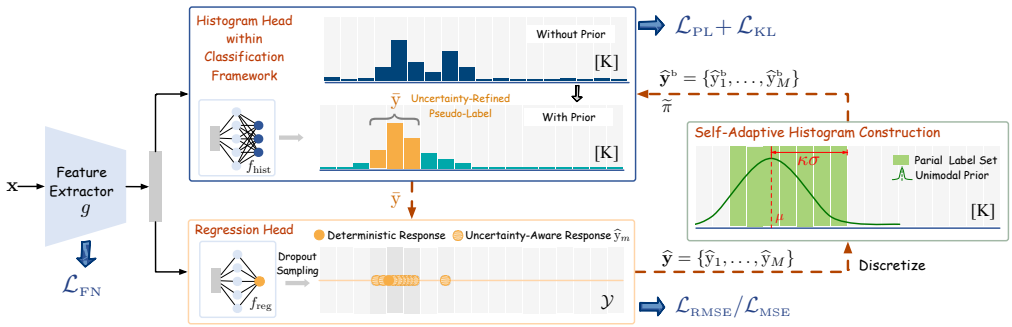


Figure 2: *Overview of MERCI. The regression head f_{reg} produces multiple dropout-based stochastic predictions, from which the instance-level mean μ and standard deviation σ are estimated. These statistics define a unimodal prior with variance scaling factor κ, and the stochastic prediction set $\hat{\mathbf{y}}$ is discretized into $\hat{\mathbf{y}}^b$ to supervise the histogram head. The histogram head f_{hist} refines these supervision signals via $\mathcal{L}_{\text{PL}}$ and $\mathcal{L}_{\text{KL}}$, and the resulting refined continuous pseudo-labels $\bar{\mathrm{y}}$ are fed back to guide f_{reg} through the regression losses $\mathcal{L}_{\text{RMSE}}$ or $\mathcal{L}_{\text{MSE}}$. The red dashed arrows in the figure illustrate the cross-head interactions within the MERCI framework.*

4.1 Regression-to-Classification: Histogram-Based Posterior Estimation

We begin by describing how to leverage the regression model's predictions to guide the training of the histogram head and obtain a discrete approximation of the conditional distribution $\mathbb{P}(\mathrm{Y}^{\mathrm{T}}|\mathrm{X}^{\mathrm{T}} = \mathrm{x})$. Specifically, let f_{hist} denote the histogram head applied after the shared feature extractor g, such that $h_{\text{hist}} \triangleq f_{\text{hist}} \circ g : \mathcal{X} \to \Delta^{K-1}$ maps each input to a K-way discrete probability distribution, which we refer to as a *histogram* throughout the paper. Here, the label space $\mathcal{Y}$ is uniformly partitioned into K bins of equal width: $\mathcal{Y} = \cup_{k=1}^{K} \mathcal{Y}_k$, where the number of bins K is chosen adaptively in our implementation ~~(see Section C.1 for details).~~ Under this binning scheme, the kth element of the histogram head output, $h_{\text{hist}}(\mathrm{x})[k]$, approximates the probability that the target label falls within bin

$\mathcal{Y}_k$, i.e., $\mathbb{P}(\mathrm{Y}^{\mathrm{T}} \in \mathcal{Y}_k | \mathrm{X}^{\mathrm{T}} = \mathrm{x})$. For each bin $\mathcal{Y}_k$, we associate a representative value $\widetilde{\mathrm{y}}_k$, such as the left endpoint, right endpoint, or midpoint of the interval. We denote the collection of all representative values as $\widetilde{\mathcal{Y}} \triangleq \{\widetilde{\mathrm{y}}_1, \ldots, \widetilde{\mathrm{y}}_K\}$, with each $\widetilde{\mathrm{y}}_k$ corresponding uniquely to a class index (i.e., bin index) k in the discrete label space $[K]$. We have included the details of selecting the bin number K and assigning the bin representative values in the Appendix C.1.

Histogram Approximation. To capture predictive uncertainty for a given input x from the target domain, we perform M stochastic forward passes through the source regression model by enabling dropout at inference time. This yields an *ensemble of predictions* $\widehat{\mathbf{y}} \triangleq \{\widehat{\mathrm{y}}_1, \ldots, \widehat{\mathrm{y}}_M\}$, which are used to construct a partial label set to supervise the learning of the histogram head. To further refine the approximation of the target label distribution, we incorporate structural characteristics inherent to many regression tasks. In many real-world settings, such as age estimation, house price prediction, or depth/angle estimation, the label distribution conditioned on a given input tends to be unimodal, with the target value concentrated around a plausible mean and exhibiting smooth variation. To reflect this, we introduce a *unimodal prior distribution* over the label space $\mathcal{Y}$, denoted as $\pi(\mathrm{y}|\mathrm{x})$. For simplicity, in our implementation, we construct a Gaussian prior for π, as detailed in Section C.1.

To integrate the unimodal prior into the histogram-based approximation, we compute the probability mass of π over the K predefined bins and use these values as a discrete prior distribution on $\widetilde{\mathcal{Y}}$, denoted as $\widetilde{\pi}$. Additionally, let $\widehat{\mathbf{y}}^{\mathrm{b}} \triangleq \{\widehat{\mathrm{y}}^{\mathrm{b}}_1, \ldots, \widehat{\mathrm{y}}^{\mathrm{b}}_M\}$ denote the discrete label set obtained by mapping the original continuous predictions $\widehat{\mathbf{y}}$ to their corresponding bin indices. By combining the uncertainty-aware partial label set $\widehat{\mathbf{y}}^{\mathrm{b}}$ with the unimodal prior belief $\widetilde{\pi}$, we approximate the histogram distribution for the unknown target label Y^{T} through the following optimization problem:

$$\widetilde{p}^{\star} \in \operatorname*{arg\,inf}_{\widetilde{p} \in \Pi} \left\{ \ell_{\mathrm{PL}}(\widetilde{p}; \widehat{\mathbf{y}}^{\mathrm{b}}) + \lambda_{\mathrm{prior}} \mathtt{d}(\widetilde{p}, \widetilde{\pi}) \right\}. \tag{1}$$

Here, $\Pi \subseteq \Delta^{K-1}$ denotes a parameterized set of discrete distributions over K classes, induced by the hypothesis space of the histogram model h_{hist}, and λ_{prior} is a learnable coefficient. For ease of presentation, we assume that Π is a convex and closed set; otherwise, we replace it with its closed convex hull in (1), as defined in Definition B.4 and further discussed in Remark B.1. The function $\ell_{\mathrm{PL}}(\widetilde{p}; \widehat{\mathbf{y}}^{\mathrm{b}})$ represents the loss associated with the partial label set $\widehat{\mathbf{y}}^{\mathrm{b}}$, and we assume that the mapping $\widetilde{p} \mapsto \ell_{\mathrm{PL}}(\widetilde{p}; \widehat{\mathbf{y}}^{\mathrm{b}})$ is convex and lower semicontinuous; in our implementation, we instantiate ℓ_{PL} using the partial label loss defined in (C3). The divergence function $\mathtt{d}(\cdot, \cdot)$ is defined in Definition B.1.

Interpretation. The optimization problem in (1) admits a twofold interpretation. First, it can be viewed as a generalization of Bayesian inference, characterized by three key arguments: (a) a loss function ℓ_{PL}, (b) a divergence $\mathtt{d}$ measuring the deviation from a prior $\widetilde{\pi}$, and (c) a feasible solution space Π. From this perspective, the optimal solution $\widetilde{p}^{\star}$ to (1) can be interpreted as an extension to the generalized variational Bayesian posterior (Knoblauch et al., 2019; Husain & Knoblauch, 2022; Soen et al., 2024), balancing empirical performance with adherence to prior knowledge. The Bayesian interpretation of (1) is further discussed in Appendix B.2.

Beyond the Bayesian viewpoint, we also establish a robustness property associated with $\widetilde{p}^{\star}$, highlighting its resilience to certain perturbations. We formalize this connection in the following proposition.

Proposition 4.1. *Let $\mathtt{d}^*_{\widetilde{\pi}}(\cdot)$ denote the the Legendre-Fenchel conjugate of $\mathtt{d}(\cdot, \widetilde{\pi})$, as defined in Definition B.3. Let $\ell^{\star}$ represent a maximizer of the optimization problem:*

$$\ell^{\star} \in \operatorname*{arg\,sup}_{\ell \in \mathcal{F}_b([K])} \left\{ \inf_{\widetilde{p} \in \Pi} \left[\ell_{\mathrm{PL}}(\widetilde{p}; \widehat{\mathbf{y}}^{\mathrm{b}}) + \mathbb{E}_{\widetilde{p}}\{\ell(\mathrm{Y}^{\mathrm{T}})\} \right] - \lambda_{\mathrm{prior}} \mathtt{d}^*_{\widetilde{\pi}}\Big(\frac{\ell}{\lambda_{\mathrm{prior}}}\Big) \right\}, \tag{2}$$

where $\mathcal{F}_b([K])$ denotes the set of all bounded and measurable functions mapping from the discrete label space $[K]$ to $\mathbb{R}$. Then, for the optimal solution $\widetilde{p}^{\star}$ to (1), the following holds:

$$\widetilde{p}^{\star} \in \operatorname*{arg\,inf}_{\widetilde{p} \in \Pi} \left[\ell_{\mathrm{PL}}(\widetilde{p}; \widehat{\mathbf{y}}^{\mathrm{b}}) + \mathbb{E}_{\widetilde{p}} \{\ell^{\star}(\mathrm{Y}^{\mathrm{T}})\} \right]. \tag{3}$$

The proof of Proposition 4.1 is provided in Appendix B.2. According to Proposition 4.1, the optimal solution $\widetilde{p}^{\star}$ to (1) minimizes the loss under perturbations introduced by an adversary. This implies that solving (1) to approximate the histogram distribution yields the optimal belief even when the original loss function ℓ_{PL} is replaced by a perturbed version, thereby ensuring robustness to potential distribution shift and model misspecification in the SFDA setting.

4.2 CLASSIFICATION-TO-REGRESSION: HISTOGRAM INFORMATION TRANSFER

We next describe two strategies that make use of the approximated histogram distribution to supervise the learning of the regression head.

Pseudo-Label Calculation. For each $\mathrm{x}_i^{\mathrm{T}}$ in the target domain data $\mathcal{D}_{\mathrm{T}}$, let $\widetilde{p}_i^{\text{old}} \in \Delta^{K-1}$ denote the approximated histogram distribution produced by the histogram head h_{hist}. To better stabilize the adaptation process and retain useful information from the source model's predictions (Qiu et al., 2021), we update $\widetilde{p}_i^{\text{old}}$ using a moving average strategy: $\widetilde{p}_i^{\text{old}} = \texttt{NORMALIZE}\{\alpha\widetilde{p}_i^{\text{old}} + (1-\alpha)\widetilde{p}_i\}$. To obtain a more confident pseudo-label, we compute a truncated expectation over the most probable bins in $\widetilde{p}_i^{\text{old}}$, effectively filtering out uncertain predictions. Specifically, we start from the bin with the highest predicted probability and sequentially include bins in descending order of probability until the cumulative mass reaches a predefined confidence mass threshold τ. Let $\mathcal{I}_i \subseteq [K]$ denote the minimal such subset of bins, formally defined as:

$$\mathcal{I}_i \in \underset{\mathcal{S}\subseteq[K]}{\arg\min} |\mathcal{S}| \;\text{ s.t. }\; \sum_{j\in\mathcal{S}} \widetilde{p}_i^{\text{old}}[j] \geq \tau.$$

The continuous pseudo-label is then computed as a normalized weighted average over the representative values $\widetilde{\mathrm{y}}_j$ corresponding to the selected bins:

$$\bar{\mathrm{y}}_i \triangleq \frac{1}{\mathtt{C}} \sum_{j\in\mathcal{I}_i} \widetilde{p}_i^{\text{old}}[j] \cdot \widetilde{\mathrm{y}}_j, \text{ where } \mathtt{C} = \sum_{j\in\mathcal{I}_i} \widetilde{p}_i^{\text{old}}[j]. \tag{4}$$

During training, these uncertainty-refined pseudo-labels are used to supervise the learning of the regression head by minimizing the following batch-wise mean squared error (MSE) over a batch $\mathcal{B}$:

$$\mathcal{L}_{\text{MSE}} \triangleq \frac{1}{|\mathcal{B}|} \sum_{i\in\mathcal{B}} \{h_{\text{reg}}(\mathrm{x}_i^{\mathrm{T}}) - \bar{\mathrm{y}}_i^{\mathrm{T}}\}^2. \tag{5}$$

Robust Loss Version for Linear Regression. In addition to the standard MSE loss, we further account for the uncertainty inherent in the pseudo-labels, which is an important consideration in the SFDA setting due to potential domain shifts. To this end, we adopt a distributionally robust optimization (DRO) objective that minimizes the worst-case risk over a neighborhood of the empirical distribution. Specifically, let $\mathrm{z}_i^{\mathrm{T}} \triangleq g(\mathrm{x}_i^{\mathrm{T}})$ denote the feature representation of input $\mathrm{x}_i^{\mathrm{T}}$ obtained from the shared extractor. We express the regression head f_{reg} as a linear function: $y = \langle \beta, z^{\mathrm{T}} \rangle$, where $\langle \cdot, \cdot \rangle$ denotes the inner product, the weight vector β incorporates both the weights and bias term (with the bias absorbed by appending a constant to z), and we define the pointwise MSE as: $\ell_{\text{MSE}}(\mathrm{z}, \mathrm{y}; \beta) = (\langle \beta, \mathrm{z} \rangle - \mathrm{y})^2$. Let $\widehat{p}_{N_{\mathrm{T}}}$ denote the empirical distribution over the pseudo-label-feature pairs $\{(\mathrm{z}_i^{\mathrm{T}}, \bar{\mathrm{y}}_i)\}_{i=1}^{N_{\mathrm{T}}}$. To incorporate pseudo-label uncertainty, we minimize the following worst-case expected loss over all distributions within a divergence ball of radius δ centered at $\widehat{p}_{N_{\mathrm{T}}}$:

$$\inf_{\beta\in\mathbb{R}^r} \sup_{p:\, \mathtt{W}_2(p,\, \widehat{p}_{N_{\mathrm{T}}})\leq\delta} \mathbb{E}_p\{\ell_{\text{MSE}}(\mathrm{Z}^{\mathrm{T}}, \bar{\mathrm{Y}}^{\mathrm{T}}; \beta)\}, \tag{6}$$

where r denotes the feature dimension, $\mathtt{W}_2(\cdot,\cdot)$ represents the optimal transport divergence with square cost function, as defined in Definition B.2, and the expectation in (6) is taken with respect to the joint distribution of $(\mathrm{Z}^{\mathrm{T}}, \bar{\mathrm{Y}}^{\mathrm{T}})$ induced by p.

Lemma 4.1. *(Blanchet et al., 2019) The optimal solution $\beta^\star$ to the worst-case risk minimization problem in (6) can be equivalently obtained by solving the following regularized objective:*

$$\beta^\star \in \underset{\beta\in\mathbb{R}^r}{\arg\inf} \left\{ \mathcal{L}_{\text{RMSE}}(\beta; \{(\mathrm{z}_i^{\mathrm{T}}, \bar{\mathrm{y}}_i)\}_{i=1}^{N_{\mathrm{T}}}) + \sqrt{\delta}\|\beta\|_2 \right\},$$

where $\mathcal{L}_{\text{RMSE}}(\beta; \{(\mathrm{z}_i^{\mathrm{T}}, \bar{\mathrm{y}}_i)\}_{i=1}^{N_{\mathrm{T}}}) \triangleq \sqrt{\frac{1}{N_{\mathrm{T}}} \sum_{i=1}^{N_{\mathrm{T}}} \left(\langle \beta, \mathrm{z}_i \rangle - \bar{\mathrm{y}}_i\right)^2}$, referred to as root mean square error (RMSE).

As indicated by Lemma 4.1, once the pseudo-labels $\bar{\mathrm{y}}_i$ are obtained, a solution that is robust to potential uncertainty can be derived by solving a regularized risk minimization problem. Specifically, the distributionally robust solution corresponds to minimizing the RMSE with an ℓ_2 regularization

term (i.e., weight decay). Therefore, we optimize the regression head by minimizing the following batch-wise RMSE objective:

$$\mathcal{L}_{\text{RMSE}} = \sqrt{\mathcal{L}_{\text{MSE}}} = \sqrt{\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}\{h_{\text{reg}}(\mathrm{x}_i^{\mathrm{T}}) - \bar{\mathrm{y}}_i^{\mathrm{T}}\}^2}. \tag{7}$$

4.3 Overall Training Objective

For each input $\mathrm{x}_i^{\mathrm{T}}$ in a batch $\mathcal{B}$, let $\widehat{\mathbf{y}}_i^{\mathrm{b}} \triangleq \{\widehat{y}_{i,1}^{\mathrm{b}}, \ldots, \widehat{y}_{i,M}^{\mathrm{b}}\}$ denote the discrete labels obtained by mapping the continuous predictions to the associated bin indices; $\widetilde{p}_i = h_{\text{hist}}(\mathrm{x}_i^{\mathrm{T}}) \in \Delta^{K-1}$ the current histogram output; $\widetilde{p}_i^{\text{old}}$ the histogram obtained from previous training epochs; and $\widetilde{\pi}_i$ the prior constructed from $\widetilde{p}_i^{\text{old}}$ and $\widehat{\mathbf{y}}_i^{\mathrm{b}}$. We define the following batch-wise loss terms:

$$\mathcal{L}_{\text{PL}} = \tfrac{1}{|\mathcal{B}|}\textstyle\sum_{i\in\mathcal{B}} \ell_{\text{PL}}(\widetilde{p}_i; \widehat{\mathbf{y}}_i^{\mathrm{b}}), \text{ with } \ell_{\text{PL}}(\widetilde{p}_i; \widehat{\mathbf{y}}_i^{\mathrm{b}}) \triangleq -\tfrac{1}{|\widehat{\mathbf{y}}_i^{\mathrm{b}}|}\textstyle\sum_{k\in\widehat{\mathbf{y}}_i^{\mathrm{b}}} \log \widetilde{p}_i[k]\,;$$

$\mathcal{L}_{\text{KL}} = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \ell_{\text{KL}}(\widetilde{p}_i; \widehat{\mathbf{y}}_i^{\mathrm{b}}, \widetilde{p}_i^{\text{old}})$, with $\ell_{\text{KL}}(\widetilde{p}_i; \widehat{\mathbf{y}}_i^{\mathrm{b}}, \widetilde{p}_i^{\text{old}}) = \mathtt{d}_{\text{KL}}(\widetilde{p}, \widetilde{\pi}) = \sum_{k=1}^{K} \widetilde{\pi}_i[k] \cdot \log\left(\frac{\widetilde{\pi}_i[k]}{\widetilde{p}_i[k]}\right)$, where $\mathcal{L}_{\text{PL}}$ and $\mathcal{L}_{\text{KL}}$ correspond to the partial-label and the prior-regularization terms in (1), respectively. In addition, to account for the role of feature scale in regression (Chen et al., 2021), we optionally include a feature-norm consistency term: $\mathcal{L}_{\text{FN}} \triangleq \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} \big|\|g(\mathrm{x}_i^{\mathrm{T}})\| - \|g_{\mathrm{S}}(\mathrm{x}_i^{\mathrm{T}})\|\big|$. Consequently, the final objective, depending on whether $\mathcal{L}_{\text{MSE}}$ in (5) or $\mathcal{L}_{\text{RMSE}}$ in (7) is used, is given by:

$$\mathcal{L}_{\text{MERCI}} = \lambda_{\text{PL}}\mathcal{L}_{\text{PL}} + \lambda_{\text{prior}}\mathcal{L}_{\text{KL}} + \lambda_{\text{MSE}}\mathcal{L}_{\text{MSE}}\ (+\lambda_{\text{FN}}\mathcal{L}_{\text{FN}});$$
$$\mathcal{L}_{\text{MERCI-R}} = \lambda_{\text{PL}}\mathcal{L}_{\text{PL}} + \lambda_{\text{prior}}\mathcal{L}_{\text{KL}} + \lambda_{\text{RMSE}}\mathcal{L}_{\text{RMSE}}\ (+\lambda_{\text{FN}}\mathcal{L}_{\text{FN}}),$$

where λ_{PL}, λ_{prior}, $\lambda_{\text{MSE}}/\lambda_{\text{RMSE}}$, and λ_{FN} are learnable weighting coefficients.

5 Experimental Results

Datasets and Evaluation Metrics. We conduct experiments on four regression tasks: **age estimation** on UTKFace (Zhang et al., 2017), **head pose estimation** on Biwi-Kinect (Fanelli et al., 2013), **house price prediction** on the California Housing dataset (Pace & Barry, 1997), and **digit prediction** on two digit datasets, SVHN (Netzer et al., 2011) and MNIST (LeCun et al., 1998). For the UTKFace and Biwi-Kinect datasets, we use gender as the domain attribute and consider two domains, Female and Male, for adaptation. In the California Housing dataset, we treat different geographic regions (Near Bay and Far Bay) as distinct domains. For the digit datasets, we follow the setup in Adachi et al. (2024), and adaptation is performed across different datasets. To comprehensively evaluate the performance of our regression models, we employ four widely used evaluation metrics: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Coefficient of Determination (R^2), and Pearson Correlation Coefficient (R).

Methods. As a relatively underexplored research topic, Source-Free Domain Adaptation for Regression (SFDAR) lacks established baselines. To enable a comprehensive evaluation, we compare our approach with representative methods from three related settings: Unsupervised Domain Adaptation (UDA), Data-Free (DF) Domain Adaptation, and Source-Free (SF) Domain Adaptation. For the UDA setting, we adopt DANN (Ganin et al., 2016), RSD (Chen et al., 2021), and DARE-GRAM (Nejjar et al., 2023). For the data-free setting, we consider SSA (Adachi et al., 2024) and TASFAR (He et al., 2024), which require no raw source data but rely on source-derived statistics or pre-trained components. To enhance comparison within the SF setting, we additionally implement the Batch Normalization adaptation baseline (Benz et al., 2021) and introduce two augmentation-based baselines, VM and AugSelfTr, which operate without any source data or statistics (Zhang et al., 2022a). For our methods, **MERCI** and **MERCI-R** correspond to training with $\mathcal{L}_{\text{MERCI}}$ and $\mathcal{L}_{\text{MERCI-R}}$, respectively, while "**w. FN**" denotes the use of feature norm regularization ($\mathcal{L}_{\text{FN}}$).

Further details, including dataset descriptions, evaluation metric computations, implementations details, and experimental configurations, are provided in Appendix D.

Table 1: **MAE** and $\mathbf{R}^2$ on **Biwi-Kinect** dataset (Fanelli et al., 2013). Best non-Oracle SF results are **bolded**, best non-Oracle results are underlined.

		Female → Male (MAE)				Male → Female (MAE)				Female → Male (R^2)				Male → Female (R^2)			
Method	**Type**	**Pitch**	**Roll**	**Yaw**	**Mean**	**Pitch**	**Roll**	**Yaw**	**Mean**	**Pitch**	**Roll**	**Yaw**	**Mean**	**Pitch**	**Roll**	**Yaw**	**Mean**
DANN (Ganin et al., 2016)	UDA	4.6776	4.6790	3.4183	4.2583	7.0520	5.0633	3.3409	6.1521	0.9213	0.6011	0.9764	0.8329	0.8674	0.6387	0.9292	0.8118
RSD (Chen et al., 2021)	UDA	4.7010	4.7304	3.3952	4.2755	7.0438	4.9587	6.0774	6.0266	0.9217	0.6060	0.9765	0.8347	0.8586	0.6542	0.9347	0.8158
DARE-GRAM (Nejjar et al., 2023)	UDA	4.8748	4.8832	3.4692	4.4091	6.9178	5.2421	5.5510	5.9036	0.9188	0.5829	0.9753	0.8257	0.8679	0.6414	0.9439	0.8178
SSA (Adachi et al., 2024)	DF	5.0149	4.9848	3.5495	4.5164	7.7513	5.1466	6.2680	6.3886	0.9092	0.5616	0.9748	0.8152	0.8659	0.6176	0.9298	0.8045
TASFAR (He et al., 2024)	DF	5.0718	5.0165	3.7800	4.6228	8.5715	5.6320	6.1813	6.7949	0.9081	0.5346	0.9724	0.8050	0.8034	0.5647	0.9048	0.7576
Source	SF	5.1743	5.0695	3.6009	4.6149	8.7046	5.3419	6.3284	6.7916	0.9092	0.5459	0.9748	0.8100	0.8049	0.5752	0.9260	0.7687
BN-adapt (Benz et al., 2021)	SF	5.0398	5.0036	3.5446	4.5293	8.2665	5.2135	6.0984	6.5261	0.9069	0.5502	0.9751	0.8107	0.8267	0.6059	0.9332	0.7886
VM	SF	4.8403	4.9430	3.4149	4.3994	8.4844	5.1861	6.1158	6.5955	0.9148	0.5597	0.9767	0.8171	0.8124	0.6048	0.9299	0.7824
AugSelfTr	SF	4.9817	4.9731	3.4001	4.4517	7.8359	5.0880	5.4812	6.1350	0.9108	0.5586	0.9769	0.8154	0.8288	0.6231	0.9410	0.7976
MERCI	SF	4.6454	4.9101	3.4350	4.3302	7.1005	5.1010	5.0537	5.7517	0.9212	0.5642	0.9767	0.8207	0.8441	0.6218	0.9527	0.8062
MERCI w. FN	SF	4.6464	4.9037	3.4126	4.3209	**7.0032**	5.0866	4.6776	5.5892	0.9222	0.5718	0.9771	0.8237	**0.8502**	0.6281	0.9594	0.8126
MERCI-R	SF	**4.5074**	4.8546	**3.3719**	**4.2446**	7.4265	5.1209	5.1885	5.9120	**0.9249**	**0.5745**	**0.9775**	**0.8256**	0.8373	0.6235	0.9502	0.8037
MERCI-R w. FN	SF	4.5551	**4.8520**	3.3933	4.2668	7.0206	**5.0580**	**4.4793**	**5.5193**	0.9226	0.5717	**0.9775**	0.8239	0.8490	**0.6362**	**0.9630**	**0.8161**
Oracle	-	0.8474	1.0630	0.8644	0.9249	1.1487	1.3535	1.0908	1.1977	0.9973	0.9805	0.9984	0.9921	0.9963	0.9767	0.9978	0.9903

5.1 Overall Results, Analysis and Discussion

Overall Results. Comprehensive experimental results on four datasets are presented in Tables 1, 2, E.2 and E.4. Overall, our proposed method, including all variants of MERCI, consistently improves the performance of the source model across all evaluation metrics without accessing any source data. In most cases, MERCI outperforms data-free (DF) methods, which still rely on source-derived information. On certain tasks, it even surpasses UDA methods that have access to the original source data. Moreover, the performance gains are more obvious on more challenging tasks (i.e., those with higher source model MAE), demonstrating the reliability of our approach under challenging scenarios. Additional experiments validating the robustness of MERCI to varying distribution shift severities are presented in Appendix E.4.

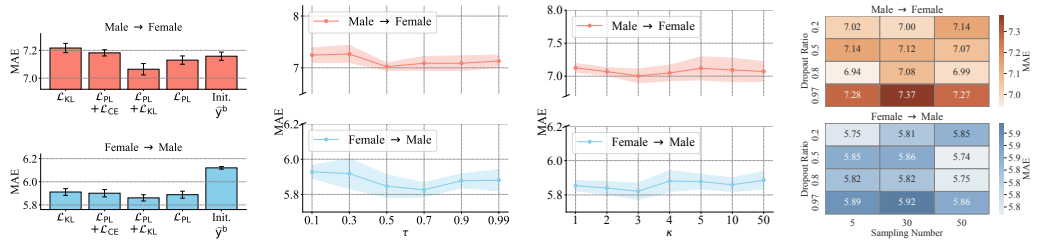
Table 2: Results on **UTKFace (Left)** (Zhang et al., 2017) and **California Housing (Right)** (Pace & Barry, 1997) datasets. Best non-Oracle SF results are **bolded**, best non-Oracle results are underlined.

		Female → Male				Male → Female				Far Bay → Near Bay				Near Bay → Far Bay			
Method	**Type**	**MAE ↓**	**RMSE ↓**	**R^2 ↑**	**R ↑**	**MAE ↓**	**RMSE ↓**	**R^2 ↑**	**R ↑**	**MAE ↓**	**RMSE ↓**	**R^2 ↑**	**R ↑**	**MAE ↓**	**RMSE ↓**	**R^2 ↑**	**R ↑**
DANN (Ganin et al., 2016)	UDA	6.2233	8.7264	0.8088	0.9005	7.0306	10.3967	0.7295	0.8623	0.4907	0.6957	0.5919	0.8129	0.5605	0.7196	0.4811	0.7365
RSD (Chen et al., 2021)	UDA	6.2172	8.7435	0.8080	0.9007	7.0332	10.3930	0.7298	0.8624	–	–	–	–	–	–	–	–
DARE-GRAM (Nejjar et al., 2023)	UDA	6.2622	8.8235	0.8045	0.8986	6.9333	10.2150	0.7390	0.8678	0.4776	0.6762	0.6144	0.8200	0.5678	0.7305	0.4647	0.7263
SSA (Adachi et al., 2024)	DF	7.0709	10.1284	0.7423	0.8861	7.6017	10.6361	0.7170	0.8530	0.4690	0.6550	0.6390	0.8017	0.7711	1.0032	-0.0065	0.4088
TASFAR (He et al., 2024)	DF	7.2788	9.7731	0.7656	0.8804	10.1191	16.6058	0.3070	0.6634	0.5055	0.6714	0.6247	0.7975	0.7366	0.9179	0.1581	0.5095
Source	SF	6.5494	9.1600	0.7892	0.8998	7.9390	12.4863	0.6098	0.8369	0.5160	0.6840	0.6050	0.7899	0.7346	0.9039	0.1830	0.5678
BN-adapt (Benz et al., 2021)	SF	6.3018	9.0182	0.7957	0.8951	7.3387	10.7872	0.7090	0.8581	0.6644	0.8767	0.3519	0.7794	0.7068	0.8840	0.2174	0.6700
VM	SF	6.0859	8.6171	0.8134	0.9060	7.4787	11.9665	0.6415	0.8373	0.4999	0.6641	0.6276	0.8065	0.6949	0.8584	0.2631	0.5940
AugSelfTr	SF	6.1548	8.6981	0.8100	0.9031	7.4219	11.8506	0.6483	0.8415	0.5092	0.6825	0.6069	0.7975	0.7108	0.8669	0.2485	0.5884
MERCI	SF	5.9570	8.3558	0.8247	0.9104	**7.0126**	10.9240	0.7013	0.8590	0.4674	0.6347	0.6602	0.8230	0.6949	0.8917	0.2047	0.6105
MERCI w. FN	SF	5.8613	8.2914	0.8273	0.9116	7.0568	11.0425	0.6949	0.8552	0.4652	0.6338	0.6612	0.8241	0.6786	0.8647	0.2520	0.6151
MERCI-R	SF	**5.8263**	**8.2325**	**0.8298**	**0.9117**	7.0802	**10.9199**	**0.7014**	0.8594	0.4661	0.6340	0.6609	0.8238	0.6909	0.8841	0.2179	0.6070
MERCI-R w. FN	SF	5.8571	8.3965	0.8229	0.9094	7.1764	11.0313	0.6954	**0.8604**	**0.4646**	**0.6290**	**0.6664**	**0.8259**	**0.6649**	**0.8503**	**0.2758**	**0.6159**
Oracle	-	5.0135	7.0991	0.8758	0.9373	4.9198	7.2291	0.8690	0.9336	0.2794	0.3947	0.8686	0.9335	0.2729	0.4115	0.8307	0.9127

Table 3: Summary of histogram information across four datasets and different adaptation tasks.

		Histogram Information		Partial Label Set Quality		Continuous Pseudo-Label Quality		
Dataset	**Task**	**Number of Bins (K)**	**Bin Length**	**Correct Label Coverage**	**Top-1 pseudo-label Accuracy**	**Source Model MAE**	**Init. $\bar{y}$ MAE**	**Best $\bar{y}$ MAE**
UTKFace	Female → Male	40.40	3.26	0.85	0.21	6.55	6.39	5.77
UTKFace	Male → Female	65.00	1.93	0.75	0.15	7.94	7.63	6.94
Biwi-Kinect	Female → Male (pitch)	36.20	3.79	0.87	0.24	5.17	4.61	4.53
Biwi-Kinect	Female → Male (yaw)	40.20	3.85	0.99	0.36	3.60	3.66	3.39
Biwi-Kinect	Female → Male (roll)	95.60	0.85	0.68	0.07	5.07	5.02	4.87
Biwi-Kinect	Male → Female (pitch)	36.00	3.88	0.69	0.14	8.70	8.74	7.40
Biwi-Kinect	Male → Female (yaw)	36.20	3.90	0.83	0.23	6.33	6.56	5.16
Biwi-Kinect	Male → Female (roll)	64.20	1.11	0.67	0.09	5.34	5.30	5.07
Digits	SVHN → MNIST	20.20	0.80	0.75	0.25	1.72	1.77	1.51
Digits	MNIST → SVHN	40.80	0.56	0.54	0.11	2.91	3.08	2.49
California House	Near Bay → Far Bay	74.00	0.26	0.97	0.13	0.73	0.71	0.66
California House	Far Bay → Near Bay	66.00	0.17	0.98	0.12	0.52	0.48	0.47

Ablation Study. We first conduct ablation experiments to investigate the different training strategies for the **histogram head** f_{hist}. As shown in Figure 3a, applying the partial label loss ($\mathcal{L}_{\text{PL}}$) improves regression performance, and combining it with the KL divergence loss ($\mathcal{L}_{\text{KL}}$) consistently yields the best results. To further assess the effectiveness of incorporating the histogram head f_{hist}, we conduct an ablation study where the regressor is directly trained using the histogram expectation estimated from the partial label set, denoted as Init. $\widehat{\mathbf{y}}^{\text{b}}$ in Figure 3a. This baseline yields unstable performance,



(a) Ablation Study on f_{hist} Training (b) Confidence Mass Threshold τ (c) Variance Scaling Factor κ (d) Dopout Configs

Figure 3: *Ablation and hyperparameter sensitivity analysis on the UTKFace dataset*

underscoring the importance of a learnable histogram head. **For the regressor**, as shown in Tables 1 and 2, MERCI-R consistently outperforms MERCI by employing a robust RMSE-based loss that is more tolerant to noisy supervision. Additionally, feature scale normalization ($\mathcal{L}_{\text{FN}}$) contributes to stabilizing MSE-based training in more challenging SFDAR setting, such as on the Biwi-Kinect (Male $\rightarrow$ Female). Further ablation results and discussions are provided in Appendix E.2.

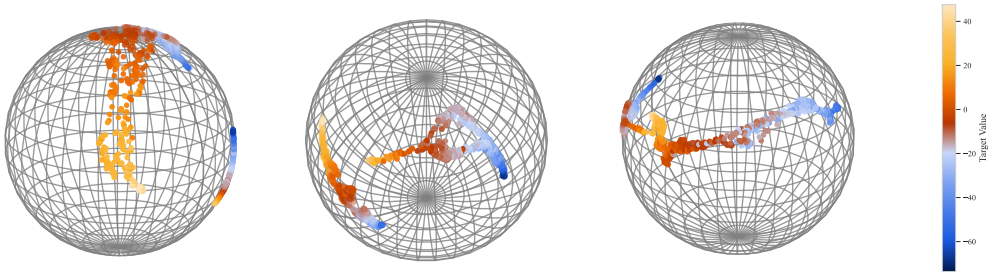
Histogram Information Analysis. To better assess the quality of the histogram and its role in correcting supervision signals, we summarize key statistics across all adaptation tasks in Table 3. The "*Histogram Information*" columns demonstrate that our self-adaptive histogram construction consistently produces reasonable bin widths and corresponding bin numbers, validating its applicability across datasets. To further understand the benefit of introducing a histogram head f_{hist} and the partial label loss, we evaluate the "*Partial Label Set Quality*" and find that ground-truth regression values are well covered by the assigned intervals, even though the top-1 pseudo-label is unreliable. This highlights that the partial label set as a whole could provide meaningful and robust supervision for label correction under weak supervision. Finally, we compare the initial and best "*Continuous Pseudo-Labels*" $\bar{\text{y}}$, taking the source model prediction as reference. The initial pseudo-labels (Init. $\bar{\text{y}}$) outperform the source model slightly but exhibit instability, whereas the best pseudo-labels (Best $\bar{\text{y}}$) obtained during adaptation are markedly more accurate, often matching or surpassing the final adapted regressor. These results highlight the effectiveness of bi-directional learning between the regression and histogram heads, and further justify the introduction of f_{hist} instead of directly supervising the regressor with the initial histogram information.

Hyperparameter Sensitivity. We perform a sensitivity analysis of four key hyperparameters: the confidence mass threshold τ in $\mathcal{L}_{\text{MSE}}$ and $\mathcal{L}_{\text{RMSE}}$, the variance scaling factor κ in $\mathcal{L}_{\text{KL}}$, and the dropout configurations—namely the dropout ratio ρ and the sampling number M. As shown in Figures 3b and 3c, MERCI maintains stable performance over a wide range of τ (0.3–0.7) and κ (1–4). We thus fix $\tau = 0.68$ and $\kappa = 3$ in all experiments. Since dropout configurations affect the initial histogram construction, we observe stable performance under moderate dropout ratios and sampling numbers, and accordingly set $\rho = 0.8$ and $M = 30$ for all datasets. For the remaining hyperparameters, we set the moving average coefficient α to 0.5 across all datasets, and treat the coefficients of different loss terms (e.g., λ_{PL}, λ_{prior}, λ_{RMSE}) as learnable parameters to simplify the selection process and better fit the SFDAR problem. Additional sensitivity analysis is provided in Appendix E.3.

Feature Representation Learning. We present UMAP visualizations (McInnes et al., 2018) of the target feature space before and after adaptation on Biwi-Kinect and UTKFace datasets. As shown in Figures 4 and E.6, source features retain partial ordering but remain loosely structured, whereas adapted features are more locally compact. Moreover, incorporating f_{hist} introduces global diversity. These observations are consistent with the criteria of a well-performing regressor (Zhang et al., 2023) and further support the effectiveness of the proposed discretized information and the histogram head. Detailed analyses are provided in Appendix E.6.

6 Conclusion

The conditional distribution of the target label is a vital yet often overlooked aspect in deep regression learning, especially in SFDAR, where obtaining such information becomes particularly challenging. In this work, we propose MERCI, a novel framework tailored for SFDAR. By leveraging a learned, sample-wise histogram, MERCI effectively captures the underlying distributional characteristics of



(a) Target Data from Source Model

(b) Target Data after Adaptation (Without f_{hist})

(c) Target Data after Adaptation (MERCI-R)

Figure 4: *UMAP visualization of target data feature space on Biwi-Kinect dataset*

target data without requiring ground-truth labels or source data. It further generates uncertainty-aware pseudo-labels through truncated histogram expectations to facilitate robust regressor adaptation. Supported by both theoretical analysis and comprehensive experiments, MERCI demonstrates strong and consistent performance across a range of regression tasks under domain shift. This work provides an insightful direction for modeling and leveraging uncertainty in regression under domain shift, paving the way for more reliable and generalizable adaptive regression framework.

Limitations and Extensions. One limitation of the current implementation lies in its reliance on sampling over the entire target dataset to construct the histogram representation. This process can become time-consuming when applied to large-scale datasets. A potential solution/future direction is to explore strategies that enable effective histogram learning based on representative data subsets, thereby improving scalability and efficiency.

Ethics Statement

We confirm that this work does not involve human subjects, animal studies, or personally identifiable data. The datasets used (e.g., UTKFace, Biwi-Kinect, California Housing, and Digits) are publicly available and widely used for academic research. All experimental protocols comply with community ethical standards, and no additional ethical approval was required.

Reproducibility Statement

We provide detailed descriptions of our implementation procedures in Section 5 and Appendices C and D to ensure transparency and reproducibility. All experiments are conducted under a unified setup with consistent architecture, optimizer, learning rate, batch size, and training epochs to ensure fair comparison. The core algorithmic code is included in the supplementary materials, and we will publicly release dataset download instructions, the full implementation, and trained source model checkpoints.

References

Kazuki Adachi, Shin'ya Yamaguchi, Atsutoshi Kumagai, and Tomoki Hamagami. Test-time adaptation for regression by subspace alignment. *arXiv preprint arXiv:2410.03263*, 2024.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Revisiting batch normalization for improving corruption robustness. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 494–503, 2021.

Jose Blanchet, Yang Kang, and Karthyek Murthy. Robust wasserstein profile inference and applications to machine learning. *Journal of Applied Probability*, 56(3):830–857, 2019.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

T Tony Cai and Hongji Wei. Transfer learning for nonparametric classification. *The Annals of Statistics*, 49(1):100–128, 2021.

Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3174–3182, 2017.

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

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

Xinyang Chen, Sinan Wang, Jianmin Wang, and Mingsheng Long. Representation subspace distance for domain adaptation regression. In *International Conference on Machine Learning*, pp. 1749–1759, 2021.

Younjoon Chung, Hyoungseob Park, Patrick Rim, Xiaoran Zhang, Jihe He, Ziyao Zeng, Safa Cicek, Byung-Woo Hong, James S Duncan, and Alex Wong. Eta: Energy-based test-time adaptation for depth completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6001–6012, 2025.

Gabriele Fanelli, Matthias Dantone, Juergen Gall, Andrea Fossati, and Luc Van Gool. Random forests for real time 3d face analysis. *International Journal of Computer Vision*, 101:437–458, 2013.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

Saurabh Garg, Yifan Wu, Sivaraman Balakrishnan, and Zachary Lipton. A unified view of label shift estimation. *Advances in Neural Information Processing Systems*, 33:3290–3300, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Tianlang He, Zhiqiu Xia, Jierun Chen, Haoliang Li, and S-H Gary Chan. Target-agnostic source-free domain adaptation for regression tasks. In *IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 1464–1477. IEEE, 2024.

Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

Hisham Husain and Jeremias Knoblauch. Adversarial interpretation of bayesian inference. In *International Conference on Algorithmic Learning Theory*, pp. 553–572. PMLR, 2022.

Ehsan Imani and Martha White. Improving regression performance with distributional losses. In *International Conference on Machine Learning*, pp. 2157–2166. PMLR, 2018.

Ehsan Imani, Kai Luedemann, Sam Scholnick-Hughes, Esraa Elelimy, and Martha White. Investigating the histogram loss in regression. *arXiv preprint arXiv:2402.13425*, 2024.

Md Tauhidul Islam, Zixia Zhou, Hongyi Ren, Masoud Badiei Khuzani, Daniel Kapp, James Zou, Lu Tian, Joseph C Liao, and Lei Xing. Revealing hidden patterns in deep neural network feature space continuum via manifold learning. *Nature Communications*, 14:8506, 2023.

Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7482–7491, 2018.

HyunGi Kim, Siwon Kim, Jisoo Mok, and Sungroh Yoon. Battling the non-stationarity in time series forecasting via test-time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 17868–17876, 2025.

Jaebok Kim, Khiet P Truong, Gwenn Englebienne, and Vanessa Evers. Learning spectro-temporal features with 3d cnns for speech emotion recognition. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pp. 383–388, 2017.

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. Generalized variational inference: Three arguments for deriving new posteriors. *arXiv preprint arXiv:1904.02063*, 2019.

Wouter M Kouw and Marco Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):766–785, 2019.

Stéphane Lathuilière, Pablo Mesejo, Xavier Alameda-Pineda, and Radu Horaud. A comprehensive analysis of deep regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42: 2065–2081, 2019.

Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist*, 1998.

Ju-Hyoung Lee, Joonghyuk Hahn, Hyeon-Tae Seo, Jiho Park, and Yo-Sub Han. Superst: Superficial self-training for few-shot text classification. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 15436–15447, 2024.

Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning*, pp. 6028–6039, 2020.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning*, pp. 3122–3130. PMLR, 2018.

Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer feature learning with joint distribution adaptation. In *Proceedings of the IEEE international conference on computer vision*, pp. 2200–2207, 2013.

Subha Maity, Diptavo Dutta, Jonathan Terhorst, Yuekai Sun, and Moulinath Banerjee. A linear adjustment-based approach to posterior drift in transfer learning. *Biometrika*, 111:31–50, 2024.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

Yu Mitsuzumi, Akisato Kimura, and Hisashi Kashima. Understanding and improving source-free domain adaptation from a theoretical perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28515–28524, 2024.

Ismail Nejjar, Qin Wang, and Olga Fink. Dare-gram: Unsupervised domain adaptation regression by aligning inverse gram matrices. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11744–11754, 2023.

Ismail Nejjar, Gaetan Frusque, Florent Forest, and Olga Fink. Uncertainty-guided alignment for unsupervised domain adaptation in regression. *arXiv preprint arXiv:2401.13721*, 2024.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on Deep Learning and Unsupervised Feature Learning*, volume 2011, pp. 4, 2011.

R. Kelley Pace and Ronald Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33 (3):291–297, 1997.

Zhen Qiu, Yifan Zhang, Hongbin Lin, Shuaicheng Niu, Yanxia Liu, Qing Du, and Mingkui Tan. Source-free domain adaptation via avatar prototype generation and adaptation. *arXiv preprint arXiv:2106.15326*, 2021.

Olivier Rioul. This is it: A primer on shannon's entropy and information. In *Information Theory: Poincaré Seminar 2018*, pp. 49–86. Springer, 2021.

Alexander Soen, Hisham Husain, Philip Schulz, and Vu Nguyen. Rejection via learning density ratios. *arXiv preprint arXiv:2405.18686*, 2024.

Yanru Sun, Zongxia Xie, Dongyue Chen, Emadeldeen Eldele, and Qinghua Hu. Hierarchical classification auxiliary network for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20743–20751, 2025.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5018–5027, 2017.

Youhong Wang, Yunji Liang, Hao Xu, Shaohui Jiao, and Hongkai Yu. Sqldepth: Generalizable self-supervised fine-structured monocular depth estimation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 5713–5721, 2024.

Jun Wu, Jingrui He, Sheng Wang, Kaiyu Guan, and Elizabeth Ainsworth. Distribution-informed neural networks for domain adaptation regression. *Advances in Neural Information Processing Systems*, 35:10040–10054, 2022.

Gezheng Xu, Hui Guo, Li Yi, Charles Ling, Boyu Wang, and Grace Yi. Revisiting source-free domain adaptation: a new perspective via uncertainty control. In *The Thirteenth International Conference on Learning Representations*, 2024.

Lingfeng Yang, Xiang Li, Renjie Song, Borui Zhao, Juntian Tao, Shihao Zhou, Jiajun Liang, and Jian Yang. Dynamic mlp for fine-grained image classification by leveraging geographical and temporal information. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10945–10954, 2022a.

Shiqi Yang, Joost van de Weijer, Luis Herranz, and Shangling Jui. Exploiting the intrinsic neighborhood structure for source-free domain adaptation. *Advances in Neural Information Processing Systems*, pp. 29393–29405, 2021.

Shiqi Yang, Yaxing Wang, Kai Wang, Shangling Jui, and Joost van de Weijer. Attracting and dispersing: A simple approach for source-free domain adaptation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5802–5815, 2022b.

Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation. *Advances in Neural Information Processing Systems*, 35:38629–38642, 2022a.

Shihao Zhang, Linlin Yang, Michael Bi Mi, Xiaoxu Zheng, and Angela Yao. Improving deep regression with ordinal entropy. In *The Eleventh International Conference on Learning Representations*, 2023.

Shihao Zhang, Kenji Kawaguchi, and Angela Yao. Deep regression representation learning with topology. In *International Conference on Machine Learning*, pp. 59080–59099. PMLR, 2024.

Shihao Zhang, Yuguang Yan, and Angela Yao. Improving deep regression with tightness. In *The Thirteenth International Conference on Learning Representations*, 2025.

Zhifei Zhang, Yang Song, and Hairong Qi. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5810–5818, 2017.

Ziyi Zhang, Weikai Chen, Hui Cheng, Zhen Li, Siyuan Li, Liang Lin, and Guanbin Li. Divide and contrast: Source-free domain adaptation via adaptive contrastive learning. *Advances in Neural Information Processing Systems*, 35:5137–5149, 2022b.

Zhi Zhou, Kun-Yang Yu, Lan-Zhe Guo, and Yu-Feng Li. Fully test-time adaptation for tabular data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23027–23035, 2025.

Yilun Zhu, Jianxin Zhang, Aditya Gangrade, and Clay Scott. Label noise: Ignorance is bliss. *Advances in Neural Information Processing Systems*, 37:116575–116616, 2024.

A USE OF LARGE LANGUAGE MODELS (LLMS)

We acknowledge the use of LLMs in this work solely as a general-purpose assistant tool, primarily for grammar checking, sentence polishing, and writing refinement, which did not contribute to the conception, methodology, analysis, or conclusions of this work. All substantive contributions of this submission, spanning problem formulation, idea and methodology development, theoretical analysis, experimental design and implementation, result interpretation, and manuscript preparation, are entirely attributed to the human authors.

B TECHNICAL DETAILS

Before delving into the technical details, we present a notation table that summarizes the key symbols used throughout the paper. For each entry, we provide the notation in the first column, its description in the second column, and the section where it first appears in the third column.

Table B.1: Notation used throughout the paper.

Notations	Descriptions	First appearance
X/Y/Z	Random variables of input/label/feature representation	Section 1
$\mathcal{H}(\cdot\|\cdot)$	Conditional Entropy	Section 1
$\mathcal{X} \subset \mathbb{R}^d$	Input space	Section 3
$\mathcal{Y} \subset \mathbb{R}$	Label space	Section 3
$P^{\mathrm{S}}_{\mathrm{xy}}$; $\mathcal{D}_{\mathrm{S}}$	Underlying distribution over $\mathcal{X} \times \mathcal{Y}$ related to source domain; Unavailable source domain data $\mathcal{D}_{\mathrm{S}} \triangleq \{\mathrm{x}_i^{\mathrm{S}}, \mathrm{y}_i^{\mathrm{S}}\}_{i=1}^{N_{\mathrm{S}}}$	Section 3
$P^{\mathrm{T}}_{\mathrm{xy}}$; $\mathcal{D}_{\mathrm{T}}$	Underlying distribution over $\mathcal{X} \times \mathcal{Y}$ related to target domain; Unlabeled target domain data $\mathcal{D}_{\mathrm{T}} \triangleq \{\mathrm{x}_i^{\mathrm{T}}\}_{i=1}^{N_{\mathrm{T}}}$	Section 3
$g_{\mathrm{S}}/g_{\mathrm{T}}/g$	Feature extractors of source/target/general model	Section 3
$f_{\mathrm{reg,S}}/f_{\mathrm{reg,T}}/f_{\mathrm{reg}}$	Regression heads of source/target/general model	Section 3
f_{hist}	Histogram head of MERCI model	Section 4
K	Number of histogram bins	Section 3
μ, σ, κ	mean, standard deviation, and variance scaling factor of Gaussian Prior	Section 4
$\bar{\mathrm{y}}_i$	Continuous Pseudo-Label	Section 4
$\widetilde{\mathrm{y}}_k$	Representative regression value of the k-th histogram bin	Section 4.1
$\widetilde{p}_i \in \Delta^{K-1}$	Empirical discrete distribution over K histogram bins of $\mathrm{x}_i^{\mathrm{T}}$	Section 4.1
$\widetilde{\pi}$	Discrete Prior Distribution	Section 4.1
$\widehat{\mathrm{y}}_i$	Uncertainty-aware regression prediction (continuous)	Section 4.1
$\widehat{\mathrm{y}}_i^{\mathrm{b}}, \widehat{\mathbf{y}}_i^{\mathrm{b}}$	Pseudo-histogram label (i.e., bin index) and partial bin index set for $\mathrm{x}_i^{\mathrm{T}}$	Section 4.1
$\widetilde{p}_i$	Empirical distribution over the pseudo-label-feature pairs $\{(\mathrm{z}_i^{\mathrm{T}}, \bar{\mathrm{y}}_i)\}_{i=1}^{N_{\mathrm{T}}}$	Section 4.1
ℓ, $\mathcal{L}$	Instance-level and mini-batch/dataset level loss term	Section 4.1
$\mathcal{H} = \{(\mathrm{x}_i^{\mathrm{T}}, \widetilde{p}_i^{\mathrm{old}}, \widehat{\mathbf{y}}_i^{\mathrm{b}})\}_{i=1}^{N_{\mathrm{T}}}$	Histogram information set	Section C.2
$\texttt{DROPOUT}(\cdot, \rho)$	Dropout function with ratio $\rho \in [0, 1]$	Section C.2

B.1 PRELIMINARIES

Let $\mathcal{S}$ denote a set that admits Polish topology, let $\mathcal{P}^\dagger(\mathcal{S})$ denote the set of finitely-additive measures over $\mathcal{S}$, let $\mathcal{P}(\mathcal{S}) \subseteq \mathcal{P}^\dagger(\mathcal{S})$ stand for the set of Borel probability measures supported on $\mathcal{S}$, and let $\mathcal{F}_b(\mathcal{S})$ represent the set of all bounded and measurable functions mapping from $\mathcal{S}$ to $\mathbb{R}$

We first present several useful definitions and lemmas following Husain & Knoblauch (2022).

Definition B.1 (Divergences)**.** A divergence is a function $\mathtt{d} : \mathcal{P}(\mathcal{S}) \times \mathcal{P}(\mathcal{S}) \to \mathbb{R}$, such that for any $\mu_1, \mu_2 \in \mathcal{P}(\mathcal{S})$, (1) $\mathtt{d}(\mu_1, \mu_2) \geq 0$; (2) $\mathtt{d}(\mu_1, \mu_2) = 0 \Leftrightarrow \mu_1 = \mu_2$; and (3) $\mathtt{d}$ is proper convex lower semi-continuous in its first argument.

Definition B.2 (Optimal transport divergence; Blanchet et al. (2019))**.** Let $c : \mathbb{R}^q \times \mathbb{R}^q \to [0, +\infty]$ be any lower semi-continuous function such that $c(u, u) = 0$ for every $u \in \mathbb{R}^q$. Given two probability distributions $P_1(\cdot)$ and $P_2(\cdot)$ supported on $\mathbb{R}^q$, the optimal transport divergence between P_1 and P_2, denoted by $\mathtt{W}_c(P_1, P_2)$, is defined as

$$\mathtt{W}_c(P_1, P_2) = \inf\{\mathbb{E}_\pi\{c(U_1, U_2)\} : \pi \in \mathrm{Cpl}(P_1, P_2)\},$$

where $\mathrm{Cpl}(P_1, P_2)$, sometimes called the coupling set of P_1 and P_2, comprises all probability measures on the product space $\mathbb{R}^q \times \mathbb{R}^q$ such that their marginal measures are $P_1(\cdot)$ and $P_2(\cdot)$. In particular, when taking the square cost function $c(u_1, u_2) = \|u_1 - u_2\|_2^2$ for $u_1, u_2 \in \mathbb{R}^q$, we denote $\mathtt{W}_c$ as $\mathtt{W}_2$.

Definition B.3 (Legendre-Fenchel conjugate)**.** For a given $\pi \in \mathcal{P}(\mathcal{S})$, the Legendre-Fenchel conjugate of $\mathtt{d}(\cdot, \pi) : \mathcal{P}(\mathcal{S}) \to \mathbb{R}$ is defined as below:

$$\mathtt{d}_\pi^*(\ell) = \sup_{\mu \in \mathcal{P}^\dagger(\mathcal{S})} \left\{ \int_{\mathcal{S}} \ell \, d\mu - \mathtt{d}(\mu, \pi) \right\} \text{ for all } \ell \in \mathcal{F}_b(\mathcal{S}).$$

Definition B.4 (Closed convex hull)**.** For a set Π, the *convex hull* of Π, denoted $\mathrm{co}(\Pi)$, is the smallest convex set containing Π. Equivalently, it can be defined as:

$$\mathrm{co}(\Pi) = \left\{ \sum_{i=1}^n \lambda_i \mu_i : \mu_i \in \Pi, \lambda_i \geq 0, \sum_{i=1}^n \lambda_i = 1, n \in \mathbb{N} \right\}.$$

The *closed convex hull* of Π, denoted $\overline{\mathrm{co}}(\Pi)$, is the smallest closed convex set that includes Π, and can be equivalently defined as the closure of $\mathrm{co}(\Pi)$.

Remark B.1. For ease of presentation, we assume in Equation (1) (Section 4.1) that Π—the set of distributions over $\widetilde{\mathcal{Y}}$ induced by the classification model—is convex and closed. If this assumption does not hold, we may instead work with its closed convex hull $\overline{\mathrm{co}}(\Pi)$ in (1).

Lemma B.1. *For any $\mu, \pi \in \mathcal{P}(\mathcal{S})$, the following holds:*

$$\mathtt{d}(\mu, \pi) = \sup_{\ell \in \mathcal{F}_b(\mathcal{S})} \left\{ \mathbb{E}_\mu(\ell) - \mathtt{d}_\pi^*(\ell) \right\}.$$

Lemma B.2. *Let $\ell : \mathcal{P}(\mathcal{S}) \to \mathbb{R}$ denote a convex and lower semicontinuous function. For any $\pi \in \mathcal{P}(\mathcal{S})$, divergence $\mathtt{d}$, $\lambda > 0$, and set $\Pi \subseteq \mathcal{P}(\mathcal{S})$, define a function $F : \mathcal{P}(\mathcal{S}) \times \mathcal{F}_b(\mathcal{S}) \to \mathbb{R}$ as*

$$F(p, \ell) = \ell(p) + \lambda \left\{ \mathbb{E}_p(\ell) - \mathtt{d}_\pi^*(\ell) \right\} + \iota_{\overline{co}(\Pi)}(p),$$

where $\iota_{\overline{co}(\Pi)}(p) = +\infty$ if $p \notin \overline{co}(\Pi)$, and 0 otherwise. Then, it holds that

$$\inf_{p \in \mathcal{P}(\mathcal{S})} \sup_{\ell \in \mathcal{F}_b(\mathcal{S})} F(p, \ell) = \sup_{\ell \in \mathcal{F}_b(\mathcal{S})} \inf_{p \in \mathcal{P}(\mathcal{S})} F(p, \ell).$$

Lemma B.3 (Fenchel-Young inequality)**.** *Let $\mathcal{S}$ be a complete normed vector space and let $\mathcal{S}^*$ be the dual space to $\mathcal{S}$. For a function $f : \mathcal{S} \to \mathbb{R} \cup \{-\infty, +\infty\}$, its convex conjugate is the function $f^* : \mathcal{S}^* \to \mathbb{R} \cup \{-\infty, +\infty\}$ defined as: $f^*(x^*) = \sup \{\langle x^*, x \rangle - f(x) : x \in \mathcal{S}\}$ for any $x^* \in \mathcal{S}^*$. Then, it holds for every $x \in \mathcal{S}$ and $x^* \in \mathcal{S}^*$:*

$$\langle x^*, x \rangle \leq f(x) + f^*(x^*).$$

B.2 Proof of Proposition 4.1

We now present the proof of Proposition 4.1 by modifying the proofs in Husain & Knoblauch (2022).

Proof. By definition, we have

$$\ell_{\mathrm{PL}}(\widetilde{p}^\star; \widehat{\mathbf{y}}^{\mathrm{b}}) + \mathbb{E}_{\widetilde{p}^\star} \left\{ \ell^\star(\mathrm{Y}^{\mathrm{T}}) \right\} \geq \inf_{\widetilde{p} \in \Pi} \left[\ell_{\mathrm{PL}}(\widetilde{p}; \widehat{\mathbf{y}}^{\mathrm{b}}) + \mathbb{E}_{\widetilde{p}} \left\{ \ell^\star(\mathrm{Y}^{\mathrm{T}}) \right\} \right]. \tag{B1}$$

We now prove the other direction in the following.

The optimization problem in (1) can be expressed as below:

$$
\begin{aligned}
&\inf_{\widetilde{p}\in\Pi}\left\{\ell_{\mathrm{PL}}(\widetilde{p};\widehat{\mathbf{y}}^{\mathrm{b}})+\lambda_{\mathrm{prior}}\mathtt{d}(\widetilde{p},\widetilde{\pi})\right\}\\
&=\inf_{\widetilde{p}\in\Pi}\left[\ell_{\mathrm{PL}}(\widetilde{p};\widehat{\mathbf{y}}^{\mathrm{b}})+\lambda_{\mathrm{prior}}\sup_{\ell\in\mathcal{F}_b(\mathcal{S})}\left\{\mathbb{E}_{\widetilde{p}}(\ell)-\mathtt{d}^*_{\widetilde{\pi}}(\ell)\right\}\right]\\
&=\inf_{\widetilde{p}\in\mathcal{P}(\widetilde{\mathcal{Y}})}\sup_{\ell\in\mathcal{F}_b(\mathcal{S})}\left[\ell_{\mathrm{PL}}(\widetilde{p};\widehat{\mathbf{y}}^{\mathrm{b}})+\lambda_{\mathrm{prior}}\left\{\mathbb{E}_{\widetilde{p}}(\ell)-\mathtt{d}^*_{\widetilde{\pi}}(\ell)\right\}+\iota_{\Pi}(\widetilde{p})\right]\\
&=\sup_{\ell\in\mathcal{F}_b(\mathcal{S})}\inf_{\widetilde{p}\in\mathcal{P}(\widetilde{\mathcal{Y}})}\left[\ell_{\mathrm{PL}}(\widetilde{p};\widehat{\mathbf{y}}^{\mathrm{b}})+\lambda_{\mathrm{prior}}\left\{\mathbb{E}_{\widetilde{p}}(\ell)-\mathtt{d}^*_{\widetilde{\pi}}(\ell)\right\}+\iota_{\Pi}(\widetilde{p})\right]\\
&=\sup_{\ell\in\mathcal{F}_b(\mathcal{S})}\left[\inf_{\widetilde{p}\in\mathcal{P}(\widetilde{\mathcal{Y}})}\left\{\ell_{\mathrm{PL}}(\widetilde{p};\widehat{\mathbf{y}}^{\mathrm{b}})+\lambda_{\mathrm{prior}}\mathbb{E}_{\widetilde{p}}(\ell)+\iota_{\Pi}(\widetilde{p})\right\}-\lambda_{\mathrm{prior}}\mathtt{d}^*_{\widetilde{\pi}}(\ell)\right]\\
&=\sup_{\ell\in\mathcal{F}_b(\mathcal{S})}\left[\inf_{\widetilde{p}\in\Pi}\left\{\ell_{\mathrm{PL}}(\widetilde{p};\widehat{\mathbf{y}}^{\mathrm{b}})+\lambda_{\mathrm{prior}}\mathbb{E}_{\widetilde{p}}(\ell)\right\}-\lambda_{\mathrm{prior}}\mathtt{d}^*_{\widetilde{\pi}}(\ell)\right]\\
&=\sup_{\ell'\in\mathcal{F}_b(\mathcal{S})}\left[\inf_{\widetilde{p}\in\Pi}\left\{\ell_{\mathrm{PL}}(\widetilde{p};\widehat{\mathbf{y}}^{\mathrm{b}})+\mathbb{E}_{\widetilde{p}}(\ell')\right\}-\lambda_{\mathrm{prior}}\mathtt{d}^*_{\widetilde{\pi}}\big(\frac{\ell'}{\lambda_{\mathrm{prior}}}\big)\right]
\end{aligned}
\tag{B2}
$$

where the first step follows from Lemma B.1, the third step comes from Lemma B.2 and the assumption that Π is closed and convex, and the last step holds by letting $\ell'=\lambda_{\mathrm{prior}}\ell$ and applying the definition of $\mathcal{F}_b(\mathcal{S})$.

Consequently, we further obtain that

$$
\begin{aligned}
&\inf_{\widetilde{p}\in\Pi}\left[\ell_{\mathrm{PL}}(\widetilde{p};\widehat{\mathbf{y}}^{\mathrm{b}})+\mathbb{E}_{\widetilde{p}}\left\{\ell^\star(\mathrm{Y}^{\mathrm{T}})\right\}\right]-\left[\ell_{\mathrm{PL}}(\widetilde{p}^\star;\widehat{\mathbf{y}}^{\mathrm{b}})+\mathbb{E}_{\widetilde{p}^\star}\left\{\ell^\star(\mathrm{Y}^{\mathrm{T}})\right\}\right]\\
&=\inf_{\widetilde{p}\in\Pi}\left[\ell_{\mathrm{PL}}(\widetilde{p};\widehat{\mathbf{y}}^{\mathrm{b}})+\mathbb{E}_{\widetilde{p}}\left\{\ell^\star(\mathrm{Y}^{\mathrm{T}})\right\}\right]-\lambda_{\mathrm{prior}}\mathtt{d}^*_{\widetilde{\pi}}\big(\frac{\ell^\star}{\lambda_{\mathrm{prior}}}\big)\\
&\quad+\lambda_{\mathrm{prior}}\mathtt{d}^*_{\widetilde{\pi}}\big(\frac{\ell^\star}{\lambda_{\mathrm{prior}}}\big)-\left[\ell_{\mathrm{PL}}(\widetilde{p}^\star;\widehat{\mathbf{y}}^{\mathrm{b}})+\mathbb{E}_{\widetilde{p}^\star}\left\{\ell^\star(\mathrm{Y}^{\mathrm{T}})\right\}\right]\\
&=\sup_{\ell'\in\mathcal{F}_b(\mathcal{S})}\left\{\inf_{\widetilde{p}\in\Pi}\left[\ell_{\mathrm{PL}}(\widetilde{p};\widehat{\mathbf{y}}^{\mathrm{b}})+\mathbb{E}_{\widetilde{p}}\left\{\ell'(\mathrm{Y}^{\mathrm{T}})\right\}\right]-\lambda_{\mathrm{prior}}\mathtt{d}^*_{\widetilde{\pi}}\big(\frac{\ell'}{\lambda_{\mathrm{prior}}}\big)\right\}\\
&\quad+\lambda_{\mathrm{prior}}\mathtt{d}^*_{\widetilde{\pi}}\big(\frac{\ell^\star}{\lambda_{\mathrm{prior}}}\big)-\left[\ell_{\mathrm{PL}}(\widetilde{p}^\star;\widehat{\mathbf{y}}^{\mathrm{b}})+\mathbb{E}_{\widetilde{p}^\star}\left\{\ell^\star(\mathrm{Y}^{\mathrm{T}})\right\}\right]\\
&=\inf_{\widetilde{p}\in\Pi}\left\{\ell_{\mathrm{PL}}(\widetilde{p};\widehat{\mathbf{y}}^{\mathrm{b}})+\lambda_{\mathrm{prior}}\mathtt{d}(\widetilde{p},\widetilde{\pi})\right\}+\lambda_{\mathrm{prior}}\mathtt{d}^*_{\widetilde{\pi}}\big(\frac{\ell^\star}{\lambda_{\mathrm{prior}}}\big)-\left[\ell_{\mathrm{PL}}(\widetilde{p}^\star;\widehat{\mathbf{y}}^{\mathrm{b}})+\mathbb{E}_{\widetilde{p}^\star}\left\{\ell^\star(\mathrm{Y}^{\mathrm{T}})\right\}\right]\\
&=\ell_{\mathrm{PL}}(\widetilde{p}^\star;\widehat{\mathbf{y}}^{\mathrm{b}})+\lambda_{\mathrm{prior}}\mathtt{d}(\widetilde{p}^\star,\widetilde{\pi})+\lambda_{\mathrm{prior}}\mathtt{d}^*_{\widetilde{\pi}}\big(\frac{\ell^\star}{\lambda_{\mathrm{prior}}}\big)-\left[\ell_{\mathrm{PL}}(\widetilde{p}^\star;\widehat{\mathbf{y}}^{\mathrm{b}})+\mathbb{E}_{\widetilde{p}^\star}\left\{\ell^\star(\mathrm{Y}^{\mathrm{T}})\right\}\right]\\
&=\lambda_{\mathrm{prior}}\left\{\mathtt{d}(\widetilde{p}^\star,\widetilde{\pi})+\mathtt{d}^*_{\widetilde{\pi}}\big(\frac{\ell^\star}{\lambda_{\mathrm{prior}}}\big)-\mathbb{E}_{\widetilde{p}^\star}\big(\frac{\ell^\star}{\lambda_{\mathrm{prior}}}\big)\right\}\\
&\geq 0,
\end{aligned}
\tag{B3}
$$

where the second step is due to the definition of $\ell^\star$ given in Proposition 4.1, the third step follows from (B2), the fourth step comes from the definition of $p^\star_{\mathrm{hist}}$, and the last step comes from the Fenchel-Young inequality in Lemma B.3. The proof is established by combining (B1) and (B3). □

Remark B.2. Let Θ denote the sample space, π a prior on Θ, and p_θ the likelihood model indexed by $\theta\in\Theta$. Given prior π, observed data x, and a parameterized subset $\Pi\subset\mathcal{P}(\Theta)$, the associated variational posterior (Blei et al., 2017) is defined as

$$
q_{\mathrm{VI}}\in\operatorname*{arg\,inf}_{q\in\Pi}\left[\mathbb{E}_{q(\theta)}\{-\log p_\theta(x)\}+\mathtt{d}_{\mathrm{KL}}(q,\pi)\right].
$$

By extending the negative log-likelihood function to a general loss function $\ell\in\mathcal{F}_b(\Theta)$ and generalizing $\mathtt{d}_{\mathrm{KL}}$ to any divergence $\mathtt{d}$ as defined in Definition B.1, the generalized variational posterior with $\lambda>0$ (Husain & Knoblauch, 2022) is defined as

$$
q_{\mathrm{GVI}}\in\operatorname*{arg\,inf}_{q\in\Pi}\left\{\mathbb{E}_{q(\theta)}\{\ell(\theta;x)\}+\lambda\mathtt{d}(q,\pi)\right\}.
$$

Similarly, the optimization problem (1) can be expressed as:

$$q^\star \in \arg\inf_{q\in\Pi} \left\{\ell(q;x) + \lambda \mathtt{d}(q,\pi)\right\},$$

which can be viewed as a generalization of q_{GVI}. Here, the loss function ℓ need not depend linearly on q; we only require that the mapping $q \mapsto \ell(q;x)$ is convex and lower semicontinuous.

C Algorithm

C.1 Implementation Details for Histogram Approximation

In this section, we elaborate on the implementation details of the histogram distribution approximation and summarize the complete training objective.

Self-Adaptive Histogram Construction. To discretize the continuous regression outputs into histogram bins, it is crucial to choose an appropriate bin width. On one hand, excessively wide bins may cause a large portion of the data to concentrate within a few intervals, leading to degraded learning performance or training collapse. On the other hand, overly narrow bins may result in sparse class distributions, making the associated classification task difficult and reducing predictive accuracy. To balance these two extremes, we introduce a *self-adaptive histogram binning method* that automatically adjusts the bin structure based on the distribution of the regression head's outputs.

Specifically, for each input $\mathrm{x}_i^{\mathrm{T}}$ from the target dataset $\mathcal{D}_{\mathrm{T}}$, let $\widehat{\mathbf{y}}_i \triangleq \{\widehat{\mathrm{y}}_{i,1},\ldots,\widehat{\mathrm{y}}_{i,M}\}$ denote a collection of predictions from the regression head, obtained through stochastic inference techniques such as dropout. Let $\widehat{\mathcal{Y}} \triangleq \{\widehat{\mathbf{y}}_i\}_{i=1}^{N_{\mathrm{T}}}$ denote the aggregated set of all such predictions. Then, we apply Gaussian kernel density estimation (KDE) to $\widehat{\mathcal{Y}}$, and denote the resulting density function as $\widehat{h}(\cdot)$. To estimate the central tendency of the target label distribution, we identify the peak (mode) of the estimated density as: $\mathrm{y}_{\text{peak}} \in \arg\max_{\mathrm{y}\in\mathcal{Y}} \widehat{h}(\mathrm{y})$, and let α_{peak} denote the corresponding quantile level of y_{peak} in the cumulative distribution.

The *bin width* is then defined adaptively as:

$$\mathtt{b} \triangleq Q_{\alpha_{\text{peak}}+0.05}(\widehat{h}) - Q_{\alpha_{\text{peak}}-0.05}(\widehat{h})$$

where $Q_\alpha(\widehat{h})$ denotes the α-quantile of the density function $\widehat{h}$ for $\alpha \in [0,1]$. This design uses the most concentrated 10% of the data to guide the choice of bin width, allowing the binning scheme to adapt to the local density structure of the pproximated target label distribution. Given the overall range of the regression outputs and the adaptive bin width, we compute the total *number of bins* as:

$$K \triangleq \mathtt{int}\Big\{\frac{\max(\widehat{\mathcal{Y}}) - \min(\widehat{\mathcal{Y}})}{\mathtt{b}}\Big\}. \tag{C1}$$

Finally, we assign a *representative value* $\widetilde{\mathrm{y}}_k$ for each bin $\mathcal{Y}_k$ in the partition $\mathcal{Y} = \cup_{k=1}^K \mathcal{Y}_k$, we assign a *representative value* $\widetilde{\mathrm{y}}_k$, which is determined by taking into account the overall skewness of the estimated target distribution. Specifically, let η be a small tolerance parameter. Define $\mathcal{Y}_{k,\,\text{left}}$, $\mathcal{Y}_{k,\,\text{mid}}$, and $\mathcal{Y}_{k,\,\text{right}}$ denote the left endpoint, midpoint, and right endpoint of the kth bin, respectively. Then, the representative value $\widetilde{\mathrm{y}}_k$ is set as follows:

$$\widetilde{\mathrm{y}}_k \triangleq \begin{cases} \mathcal{Y}_{k,\,\text{left}}, & \text{if } \alpha_{\text{peak}} < 0.5-\eta \quad \text{(left-skewed)} \\ \mathcal{Y}_{k,\,\text{right}}, & \text{if } \alpha_{\text{peak}} > 0.5+\eta \quad \text{(right-skewed)} \\ \mathcal{Y}_{k,\,\text{mid}}, & \text{otherwise} \quad \text{(approximately symmetric)} \end{cases} \tag{C2}$$

In our implementation, we set $\eta = 0.02$.

Partial Label Loss. After adaptively partitioning the regression label space into K bins, each continuous prediction $\widehat{\mathbf{y}}_{i,j} \in \widehat{\mathbf{y}}_i$ (for $j \in [M]$) is mapped to a bin index $\widehat{\mathrm{y}}^{\mathrm{b}}_{i,j}$. We denote the resulting discrete labels as $\widehat{\mathbf{y}}^{\mathrm{b}}_i \triangleq \{\widehat{\mathrm{y}}^{\mathrm{b}}_{i,1},\ldots,\widehat{\mathrm{y}}^{\mathrm{b}}_{i,M}\}$. Given the partial label set $\widehat{\mathbf{y}}^{\mathrm{b}}_i \subset [K]$ and the approximated

histogram distribution $\widetilde{p}_i = h_{\text{hist}}(\mathrm{x}_i^{\mathrm{T}}) \in \Delta^{K-1}$, we define the following partial label loss:

$$\ell_{\text{PL}}(\widetilde{p}_i; \widehat{\mathbf{y}}_i^{\text{b}}) \triangleq -\frac{1}{|\widehat{\mathbf{y}}_i^{\text{b}}|} \sum_{k \in \widehat{\mathbf{y}}_i^{\text{b}}} \log \widetilde{p}_i[k]. \tag{C3}$$

Then, with the partial label loss defined in (C3), for a batch $\mathcal{B}$, the batch-wise partial bin set loss is:

$$\mathcal{L}_{\text{PL}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \ell_{\text{PL}}(\widetilde{p}_i; \widehat{\mathbf{y}}_i^{\text{b}}). \tag{C4}$$

Additionally, to further accommodate the source-free setting, preserve the continuity of the regression space, and retain the uncertainty information, we also introduce a method to refine the partial label set $\widehat{\mathbf{y}}_i^{\text{b}}$ in a structured way. Specifically, we identify the index range spanned by $\widehat{\mathbf{y}}_i^{\text{b}}$ as $k_{\min} \triangleq \min \widehat{\mathbf{y}}_i^{\text{b}}$ and $k_{\max} \triangleq \max \widehat{\mathbf{y}}_i^{\text{b}}$, and treat the interval $[k_{\min}, k_{\max}]$ as the core region of label predictions for $\mathrm{x}_i^{\mathrm{T}}$. To further reflect predictive uncertainty and promote smoother label assignment, we expand this interval by a relaxation factor proportional to the total number of bins. The resulting partial label (bin) set is then defined as:

$$\widehat{\mathbf{y}}_i^{\text{b+}} \triangleq \{k \in [K] : k \in [k_{\min} - \lfloor \epsilon \cdot K \rfloor, k_{\min} + \lfloor \epsilon \cdot K \rfloor]\}.$$

where $\epsilon \in (0, 1)$ controls the extent of relaxation. This relaxed binning strategy allows the model to maintain a contiguous set of plausible labels, mitigating overconfidence and better accommodating the inherent uncertainty in the predictions. In our experiments, we set $\epsilon = 0.05$.

Unimodal Prior. For each $\mathrm{x}_i^{\mathrm{T}}$, we employ a unimodal Gaussian prior $\pi(\mathrm{y}|\mathrm{x}_i^{\mathrm{T}}) \sim \mathcal{N}(\mu_i, \sigma_i^2)$ for its label distribution, where the mean μ_i and variance σ_i^2 are constructed based on the histogram distribution obtained from the previous training iteration and the partial label set $\widehat{\mathbf{y}}_i^{\text{b}}$.

Specifically, we first determine the bin index with the highest probability in the previous iteration's histogram $\widetilde{p}_i^{\text{old}}$ (or initialize it as the empirical distribution over $\widehat{\mathbf{y}}_i^{\text{b}}$), and denote this index as $k_\mu \triangleq \arg\max_{k \in [K]} \widetilde{p}_i^{\text{old}}[k]$. The corresponding representative value in $\widetilde{\mathcal{Y}}$ is then selected as the mean: $\mu_i = \widetilde{\mathrm{y}}_{k_\mu}$. To define the variance σ_i^2, we account for the spread of the refined label set around the mode. In particular, we set $\kappa\sigma_i = \mathtt{b} \cdot \max(|k_\mu - \min \widehat{\mathbf{y}}_i^{\text{b}}|, |k_\mu - \max \widehat{\mathbf{y}}_i^{\text{b}}|)$, where $\mathtt{b}$ is the bin width and $\kappa > 0$ is a tunable variance scaling factor. This formulation ensures that the prior variance adapts to the dispersion of the discrete label distribution.

Then, the discrete prior distribution $\widetilde{\pi}_i$ on $\widetilde{\mathcal{Y}}$ is obtained by approximating the probability mass of π over the K bins:

$$\widetilde{\pi}_i[k] = \frac{1}{\mathtt{C}_i} \cdot \mathtt{b}\, \varphi(\widetilde{\mathrm{y}}_k; \mu_i, \sigma_i^2) \;\text{ for } k \in [K],$$

where $\varphi(\cdot; \mu_i, \sigma_i^2)$ denote the density function for the Gaussian distribution $\mathcal{N}(\mu_i, \sigma_i^2)$, and $\mathtt{C}_i$ is a normalizing constant to ensure $\sum_{k=1}^{K} \widetilde{\pi}_i[k] = 1$. Consequently, by taking $\mathtt{d}(\cdot, \cdot)$ as the Kullback–Leibler (KL) divergence $\mathtt{d}_{\text{KL}}(\cdot, \cdot)$, the second loss term in (1) is defined as below:

$$\ell_{\text{KL}}(\widetilde{p}_i; \widehat{\mathbf{y}}_i^{\text{b}}, \widetilde{p}_i^{\text{old}}) = \mathtt{d}_{\text{KL}}\left(\widetilde{\pi}_i, \widetilde{p}_i\right) = \sum_{k=1}^{K} \widetilde{\pi}_i[k] \cdot \log\left(\frac{\widetilde{\pi}_i[k]}{\widetilde{p}_i[k]}\right) \tag{C5}$$

The associated batch-wise loss for batch $\mathcal{B}$ is:

$$\mathcal{L}_{\text{KL}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \ell_{\text{KL}}(\widetilde{p}_i; \widehat{\mathbf{y}}_i^{\text{b}}, \widetilde{p}_i^{\text{old}}). \tag{C6}$$

Overall Training Objective. Besides the MSE-based losses in Section 4.2, considering that feature scale plays a critical role in regression tasks (Chen et al., 2021), we optionally incorporate the following batch-wise feature norm regularization term to encourage alignment between the feature norms of the target and source models, thereby promoting stable training dynamics and further guiding the learning of the regression model:

$$\mathcal{L}_{\text{FN}} \triangleq \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left| \|g(\mathrm{x}_i^{\mathrm{T}})\| - \|g_{\mathrm{S}}(\mathrm{x}_i^{\mathrm{T}})\| \right|, \tag{C7}$$

where $g_{\mathrm{S}}(\cdot)$ denotes the fixed feature extractor obtained from the pre-trained source model, and $g(\cdot)$ represents the feature extractor being trained on the target domain.

Combining the components discussed above, the overall loss takes the following form, depending on whether $\mathcal{L}_{\mathrm{MSE}}$ in (5) or $\mathcal{L}_{\mathrm{RMSE}}$ in (7) is used:

$$\mathcal{L}_{\mathrm{MERCI}} = \underbrace{\lambda_{\mathrm{PL}}\mathcal{L}_{\mathrm{PL}} + \lambda_{\mathrm{prior}}\mathcal{L}_{\mathrm{KL}}}_{\mathcal{L}_{\mathrm{hist}}} + \underbrace{\lambda_{\mathrm{MSE}}\mathcal{L}_{\mathrm{MSE}}\ (+\lambda_{\mathrm{FN}}\mathcal{L}_{\mathrm{FN}})}_{\mathcal{L}_{\mathrm{reg}}}; \tag{C8a}$$

$$\mathcal{L}_{\mathrm{MERCI-R}} = \underbrace{\lambda_{\mathrm{PL}}\mathcal{L}_{\mathrm{PL}} + \lambda_{\mathrm{prior}}\mathcal{L}_{\mathrm{KL}}}_{\mathcal{L}_{\mathrm{hist}}} + \underbrace{\lambda_{\mathrm{RMSE}}\mathcal{L}_{\mathrm{RMSE}}\ (+\lambda_{\mathrm{FN}}\mathcal{L}_{\mathrm{FN}})}_{\mathcal{L}_{\mathrm{reg-r}}}, \tag{C8b}$$

where $\mathcal{L}_{\mathrm{PL}}$, $\mathcal{L}_{\mathrm{KL}}$, and $\mathcal{L}_{\mathrm{FN}}$ are given in (C4), (C6), and (C7), respectively, and λ_{PL}, λ_{prior}, λ_{MSE}/λ_{RMSE}, and λ_{FN} are weighting coefficients for the four components. To reduce the burden of manual hyperparameter tuning, these coefficients are treated as learnable parameters during training.

C.2 Training Algorithm

In this subsection, we summarize the complete training procedure in Algorithms 1-2.

Algorithm 1: Step 1: Target Data Histogram Information Set Construction

Input: Source model $h_{\mathrm{S}} = f_{\mathrm{reg,S}} \circ g_{\mathrm{S}}$, target dataset $\mathcal{D}_{\mathrm{T}}$, dropout ratio ρ, number of sampling repetitions M.

Output: Histogram bin number K, bin values $\{\widetilde{\mathrm{y}}_1, \ldots, \widetilde{\mathrm{y}}_K\}$, target data histogram information set $\mathcal{H} = \{(\mathrm{x}_i^{\mathrm{T}}, \widetilde{p}_i^{\mathrm{old}}, \widehat{\mathbf{y}}_i^{\mathrm{b}})\}_{i=1}^{N_{\mathrm{T}}}$.

Initialize the histogram information set: $\mathcal{H} = \varnothing$;
foreach $\mathrm{x}_i^{\mathrm{T}}$ *in* $\mathcal{D}_{\mathrm{T}}$ **do**
  Compute uncertainty-aware predictions $\widehat{\mathbf{y}}_i \triangleq \{\widehat{\mathrm{y}}_{i,1}, \ldots, \widehat{\mathrm{y}}_{i,M}\}$ by performing M stochastic forward passes with dropout ratio ρ, where $\widehat{\mathrm{y}}_{i,m} = f_{\mathrm{reg,S}}(\texttt{DROPOUT}(g_{\mathrm{S}}(\mathrm{x}_i^{\mathrm{T}}), \rho))$
end foreach
Compute the number of histogram bins K adaptively using Eq. (C1) based on $\{\widehat{\mathbf{y}}_i^{\mathrm{T}}\}_{i=1}^{N_{\mathrm{T}}}$;
Compute the representative bin values $\{\widetilde{\mathrm{y}}_1, \ldots, \widetilde{\mathrm{y}}_K\}$ using Eq. (C2);
foreach *target sample* $\mathrm{x}_i^{\mathrm{T}}$ **do**
  Compute the empirical discrete distribution $\widetilde{p}_i^{\mathrm{old}}$ over K bins as: $\widetilde{p}_i^{\mathrm{old}}[k] \triangleq \frac{1}{M}\sum_{m=1}^{M} \mathbb{I}\left(\widehat{\mathrm{y}}_{i,m} \in \mathrm{Bin}_k\right)$ for $k \in [K]$;
  Construct the partial label (bin) set $\widehat{\mathbf{y}}_i^{\mathrm{b}}$ by mapping $\widehat{\mathbf{y}}_i$ to the associated bin indices;
  Store the histogram information tuple $(\mathrm{x}_i^{\mathrm{T}}, \widetilde{p}_i^{\mathrm{old}}, \widehat{\mathbf{y}}_i^{\mathrm{b}})$ for the target sample into $\mathcal{H}$.
end foreach

Algorithm 2: Step 2: Joint Adaptation Training of g, f_{reg}, and f_{hist}

Input: Target data $\mathcal{D}_{\text{T}}$, histogram information set $\mathcal{H}$, source model $h_{\text{S}} = f_{\text{S}} \circ g_{\text{S}}$, number of epochs $\mathcal{T}$, number of warm-up epochs $\mathcal{T}_{\text{warm}}$, hyper-parameter $\alpha \in (0, 1)$

Output: Adapted target model $h_{\text{T}} = f_{\text{reg,T}} \circ g_{\text{T}}$

Initialize target model h_{T} with source feature extractor and regressor: $f_{\text{reg,T}} = f_{\text{reg,S}}$, $g_{\text{T}} = g_{\text{S}}$;
Initialize histogram head f_{hist} with class number K;
Initialize λ_{PL}, λ_{prior}, λ_{MSE} (λ_{RMSE}), and λ_{FN};
for *epoch* $t = 1$ **to** $\mathcal{T}$ **do**
  foreach *mini-batch* $\mathcal{B}$ *from* $\mathcal{D}_{\text{T}}$ **do**
    For each $\mathrm{x}_i^{\text{T}} \in \mathcal{B}$, calculate the histogram prediction $\widetilde{p}_i = h_{\text{hist}}(\mathrm{x}_i^{\text{T}})$;
    `// Histogram loss`
    Calculate batch-wise partial label loss $\mathcal{L}_{\text{PL}}$ by Eq (C3)-(C4);
    Calculate batch-wise KL divergence with the unimodal prior $\mathcal{L}_{\text{KL}}$ by Eq (C5)-(C6);
    Calculate batch-wise histogram loss: $\mathcal{L}_{\text{hist}} = \lambda_{\text{PL}}\mathcal{L}_{\text{PL}} + \lambda_{\text{prior}}\mathcal{L}_{\text{KL}}$;
    if $t \leq \mathcal{T}_{warm}$ **then**
      Update h_{hist}, λ_{PL}, and λ_{prior} by minimizing $\mathcal{L}_{\text{hist}}$
    else
      `// Regression loss`
      For each $\mathrm{x}_i^{\text{T}} \in \mathcal{B}$, calculate pseudo-label $\bar{\mathrm{y}}_i$ based on $\widetilde{p}_i^{\text{old}}$ using Eq. (4);
      Calculate batch-wise MSE $\mathcal{L}_{\text{MSE}}$ by (5), or batch-wise RMSE $\mathcal{L}_{\text{RMSE}} = \sqrt{\mathcal{L}_{\text{MSE}}}$;
      Calculate batch-wise feature norm loss $\mathcal{L}_{\text{FN}}$ by (C7);
      Calculate batch-wise regression loss: $\mathcal{L}_{\text{reg}} = \lambda_{\text{MSE}}\mathcal{L}_{\text{MSE}}$ $(+\lambda_{\text{FN}}\mathcal{L}_{\text{FN}})$, or $\mathcal{L}_{\text{reg-r}} = \lambda_{\text{RMSE}}\mathcal{L}_{\text{RMSE}}$ $(+\lambda_{\text{FN}}\mathcal{L}_{\text{FN}})$;
      `// Overall batch-wise loss for MERCI (MERCI-R)`
      Calculate $\mathcal{L}_{\text{MERCI}}$ ($\mathcal{L}_{\text{MERCI-R}}$) by (C8);
      Update g_{T}, $f_{\text{reg,T}}$, f_{hist}, λ_{PL}, λ_{prior}, λ_{MSE} (λ_{RMSE}), and λ_{FN} by minimizing $\mathcal{L}_{\text{MERCI}}$ ($\mathcal{L}_{\text{MERCI-R}}$) ;
    end
  end foreach
  Update $\widetilde{p}_i^{\text{old}}$ in $\mathcal{H}$: $\widetilde{p}_i^{\text{old}} = \texttt{NORMALIZE}\{\alpha\widetilde{p}_i^{\text{old}} + (1-\alpha)\widetilde{p}_i\}$;
end for

D Additional Experimental Details

D.1 Details of Datasets

UTKFace (Zhang et al., 2017) is a human face dataset used for age estimation, consisting of 24,106 images with age labels ranging from 1 to 106. Each image is additionally annotated with gender and race information. We take gender as the domain attribute and conduct two domain adaptation tasks: Female $\rightarrow$ Male and Male $\rightarrow$ Female.

Biwi-Kinect (Fanelli et al., 2013) is a 3D face dataset designed for head pose estimation, comprising 15,678 images. Following the setup in previous work (Adachi et al., 2024), we use gender as the domain attribute and train three separate models to predict pitch, yaw, and roll angles, respectively. The label ranges are $[-70°, 80°]$ for yaw, $[-70°, 70°]$ for roll, and $[-90°, 60°]$ for pitch.

California Housing (Pace & Barry, 1997) is a tabular dataset used for house price prediction. Following prior work (He et al., 2024; Adachi et al., 2024), we define the source and target domains based on geographic regions (non-coastal vs. coastal). We conduct two domain adaptation tasks: Near Bay $\rightarrow$ Far Bay and Far Bay $\rightarrow$ Near Bay. The model architecture on California Housing consists of a five-layer MLP with one BatchNorm layer and ReLU activation functions.

Digits dataset includes two widely-used digit recognition datasets, **SVHN** (Netzer et al., 2011) and **MNIST** (LeCun et al., 1998). Although originally developed for classification, we follow the setup in Adachi et al. (2024) and train a regression model to directly predict the scalar value corresponding to each digit image.

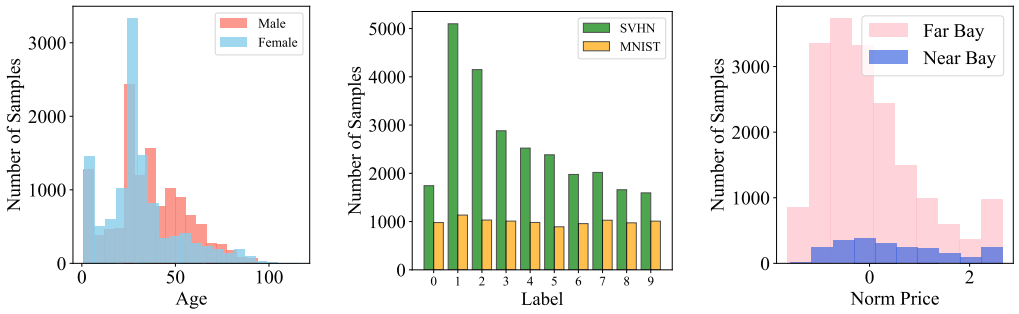
The training and validation split ratio is set following Adachi et al. (2024).

D.2 Details of SFDAR Problem

Distribution Shift Types. In SFDA settings, the problem is typically formulated under joint distribution shift, where $P^{\mathrm{S}}_{\mathrm{xy}} \neq P^{\mathrm{T}}_{\mathrm{xy}}$. This assumption encompasses covariate shift (Ben-David et al., 2006; Mansour et al., 2009; Ben-David et al., 2010), label shift (Lipton et al., 2018; Garg et al., 2020), and posterior (or concept) shift (Cai & Wei, 2021; Zhu et al., 2024; Maity et al., 2024), aligning with classical domain adaptation theory (Kouw & Loog, 2019) and joint alignment approaches (Long et al., 2013).

Heuristically guided by the decomposition $P_{\mathrm{xy}} = P_{\mathrm{x}} P_{\mathrm{y|x}}$, current SFDA research often focuses on learning more robust representations or designing stronger supervisory signals for target adaptation (Liang et al., 2020), implicitly relying on two assumptions: (1) Similarity of feature extractors: the source-trained feature extractor produces semantically meaningful representations for target inputs (Yang et al., 2022b; Liang et al., 2020); (2) Proximity of conditional distributions: $P^{\mathrm{S}}_{\mathrm{y|x}}$ and $P^{\mathrm{T}}_{\mathrm{y|x}}$ are assumed to be different but not too far apart, which facilitates the use of output uncertainty for sample selection (Zhang et al., 2022b; Xu et al., 2024).

Our method is developed under the above assumptions, with experiments conducted under a mixture of relaxed covariate (input) shift and label shift. To illustrate the range and distribution of the label space, Figures D.1 and D.2 present histograms of the ground-truth regression values across four datasets. These results demonstrate the diversity of label space distributions in the selected datasets, underscoring both the variety of experimental settings and the generalizability of the proposed method.



(a) UTKFace (b) Digits Datasets (c) California Housing

Figure D.1: *Histograms of Regression Labels for Different Domains Across Three Datasets*

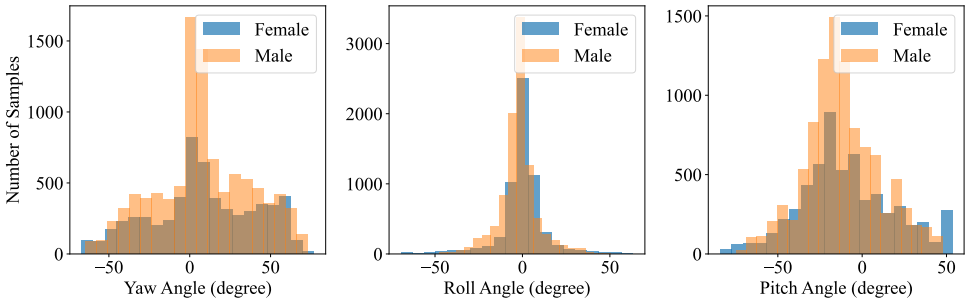


Figure D.2: *Histograms of Regression Labels for Different Domains on Biwi-Kinect Dataset*

Importance and potential application scenarios of SFDAR. Here, we aim to highlight the importance and practical relevance of the SFDAR problem. In fact, Regression is a fundamental problem in machine learning and is closely related to numerous real-world applications, including object localization, image registration, and human pose estimation (Lathuilière et al., 2019).

While SFDA has been widely studied in classification and segmentation tasks, extending this paradigm to regression is both natural and necessary. For example, (1) in healthcare, tasks like predicting

patient risk scores or tumor progression from imaging data often involve continuous outcomes. In a new hospital setting, source data may be inaccessible due to privacy concerns, and acquiring target labels requires expert annotations or follow-up exams, making SFDAR particularly relevant. (2)In industrial applications, estimating the Remaining Useful Life (RUL) of machinery is a regression task where source data is often unavailable due to commercial confidentiality, and collecting target labels is costly or impractical, as it requires observing actual failure events.

The limited prior work underscores the methodological difficulty of this setting, which makes it both challenging and valuable to investigate. By formally defining the SFDAR setting and proposing concrete solutions, we aim to encourage broader interest and further exploration in this important yet underexplored domain.

D.3 Evaluation Metrics

In this subsection, we introduce the detailed calculation of four most widely utilized regression evaluation metrics as follows:

MAE measures the average magnitude of absolute errors between predictions $y_{i,\text{pred}}$ and ground truth $y_{i,\text{true}}$:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_{i,\text{true}} - y_{i,\text{pred}}|$$

It provides an intuitive and direct estimate of the typical prediction error.

Compared to MAE, RMSE penalizes larger errors more heavily by squaring the residuals before averaging and more sensitive to outliers:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{i,\text{true}} - y_{i,\text{pred}})^2}.$$

The coefficient of determination, R^2, evaluates the proportion of variance in the target variable that is explained by the model's predictions:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_{i,\text{true}} - y_{i,\text{pred}})^2}{\sum_{i=1}^{n}(y_{i,\text{true}} - \bar{y}_{\text{true}})^2}$$

where $\bar{y}_{\text{true}}$ is the mean of the ground-truth values. A larger value means better performance and 1 indicates a perfect fit.

Finally, the Pearson correlation coefficient, R, measures the linear correlation between the predicted and true values:

$$R = \frac{\sum_{i=1}^{n}(y_{i,\text{true}} - \bar{y}_{\text{true}})(y_{i,\text{pred}} - \bar{y}_{\text{pred}})}{\sqrt{\sum_{i=1}^{n}(y_{i,\text{true}} - \bar{y}_{\text{true}})^2}\sqrt{\sum_{i=1}^{n}(y_{i,\text{pred}} - \bar{y}_{\text{pred}})^2}}$$

Values of R range from -1 (perfect negative correlation) to 1 (perfect positive correlation), with 0 indicating no linear relationship.

MAE and RMSE are error-based metrics, while R^2 and R are correlation-based metrics. Together, these metrics provide a comprehensive assessment of both prediction accuracy and model reliability.

D.4 Implementation Details

We use ResNet-26 (He et al., 2016) as the backbone for the Digits dataset, ResNet-50 for UTKFace and Biwi-Kinect, and a Multi-Layer Perceptron (MLP) for California Housing. A linear layer is used as the regression head, while MERCI adopts a bottleneck classification neural network with a 512-dimensional hidden layer for histogram learning (Liang et al., 2020). Following prior protocols, we train the source model for 100 epochs in both data-free and source-free settings. During adaptation, except for SSA, we perform full-parameter training for 30 epochs with a batch size of 64 using Adam. A smaller learning rate (0.0005) is applied to the feature extractor, and a larger one (0.005) to the regressor and classifier. In our method, loss coefficients are treated as learnable parameters and optimized using Adam with a learning rate of 0.005. All experiments are conducted using three random seeds on a single NVIDIA A100 GPU.

D.5 Details of Baseline Methods

We introduce the implementation details of different baselines as follows:

DANN (Ganin et al., 2016) is one of the most classical domain adaptation method, designed for classification task but can be directly transferred into regression. By introducing a domain classifier and the adversarial training strategy, it encourage domain-invariant feature learning, thereby aligning source and target distributions in the latent space. It jointly optimizes a label predictor and a domain discriminator through a gradient reversal layer.

RSD (Chen et al., 2021) and **DARE-GRAM** (Nejjar et al., 2023) are two unsupervised domain adaptation methods for regression that rely on representation space alignment and require simultaneous access to both source and target data.

SSA (Adachi et al., 2024) is a recently proposed test-time adaptation method based on feature alignment. Although SSA does not require access to raw source data during adaptation, it still relies on source data statistics (e.g., feature covariance matrices), which deviates from our source-free setting. We refer to this setting as a data-free domain adaptation task. Comparing the results in our setting with those reported in SSA (Adachi et al., 2024), we observe that SSA performs substantially better under covariate-shift scenarios and in shift-aware application settings. However, when facing mixed distribution shifts and changes in prediction scale, its performance tends to become less stable.

TASFAR (He et al., 2024) is another recently proposed data-free domain adaptation method designed for regression. It estimates sample-wise uncertainty through random sampling and refines regression labels accordingly. However, the method relies on the assumption of a specific distribution (e.g., Gaussian), and requires training an uncertainty predictor on source data. While TASFAR does not access raw source samples during adaptation, it still depends on source-trained components, making it incompatible with the strict source-free setting. One of its main limitations is the TASFAR's strong dependence on the quality of pseudo-labels, which may lead to performance instability, especially when the initial label estimation is inaccurate.

BN-adapt (Benz et al., 2021) is a widely used and effective source-free or test-time adaptation method. It is lightweight and well-suited for domain-adaptive regression tasks. It updates only the Batch Normalization layers' running statistics and does not require any backpropagation. Our experimental results confirm its strong performance while also revealing that BN-adapt can be sensitive to the scale of the target response space.

To further validate the effectiveness of our approach, we propose two additional data augmentation-based SFDAR methods, VM and AugSelfTr. Inspired by prior works (Adachi et al., 2024; Zhang et al., 2022a), both methods aim to minimize the variance of model outputs under data augmentation. Specifically, we apply pre-generated augmentations (AugMix (Hendrycks et al., 2019)) in **AugSelfTr**, and on-the-fly augmentations in **VM** (Chen et al., 2020).

Implementation Details of Baseline Models. For UDA and DF methods, we conduct experiments using their officially released code. For UDA methods, we perform adaptation training for 50 epochs. We briefly tune the hyperparameters for each method and report the best-performing results. Specifically, we adjust `rsd_coef` and `bmp_coef` for RSD, and `threshold`, `scale_coef` and `angle_coef` for DARE-GRAM. For the test-time adaptation method SSA, we follow the original protocol by forwarding the entire dataset once and updating only the BatchNorm layers, which yields optimal performance. For TASFAR, VM, and AugSelfTr, we train all parameters for 30 epochs to ensure a fair comparison.

E Additional Experimental Results

E.1 Additional Regression Results

Due to space limitations in the main text, we present the results of four regression metrics on the SVHN→MNIST and MNIST→SVHN adaptation tasks from the Digits dataset in Figure E.2. Additionally, the **RMSE** and R scores on the Biwi-Kinect dataset are reported in Tables E.3 and E.4.

Table E.2: Results on **Digits** datasets. Best non-Oracle SF results are **bolded**, best non-Oracle results are underlined.

Method	Type	MNIST → SVHN				SVHN → MNIST			
		MAE ↓	RMSE ↓	R^2 ↑	R ↑	MAE ↓	RMSE ↓	R^2 ↑	R ↑
DANN (Ganin et al., 2016)	UDA	2.2290	2.8487	-0.1330	0.1974	1.0980	1.7586	0.6308	0.8025
RSD (Chen et al., 2021)	UDA	2.2054	2.8162	-0.1067	0.2329	1.0528	1.7917	0.6157	0.7950
DARE-GRAM (Nejjar et al., 2023)	UDA	2.1480	2.7968	-0.0914	0.2680	1.1707	1.8072	0.6105	0.7853
SSA (Adachi et al., 2024)	DF	2.6255	3.5262	-0.7360	0.1364	1.3734	2.1633	0.4395	0.6911
TASFAR (He et al., 2024)	DF	2.8918	3.7616	-0.9969	0.0845	1.6768	2.2270	0.4024	0.6426
Source	SF	2.9124	3.7935	-1.0305	0.1043	1.7155	2.3422	0.3419	0.5951
BN-adapt (Benz et al., 2021)	SF	2.5836	3.4625	-0.6732	0.1335	1.4787	2.2426	0.3955	0.6465
VM	SF	2.8228	3.7007	-0.9343	0.1213	1.6513	2.2604	0.3849	0.6246
AugSelfTr	SF	2.8423	3.7073	-0.9436	0.1178	1.5995	2.1078	0.4632	0.6897
MERCI	SF	**2.4378**	**3.2164**	**-0.4590**	0.1623	**1.4770**	**1.9750**	**0.5290**	**0.7665**
MERCI w. FN	SF	2.4593	3.2423	-0.4830	**0.1814**	1.4971	1.9982	0.5183	0.7481
MERCI-R	SF	2.4808	3.2591	-0.4952	0.1411	1.5090	1.9852	0.5172	0.7376
MERCI-R w. FN	SF	2.5207	3.3101	-0.5447	0.1698	1.5159	2.0138	0.5100	0.7534
Oracle	-	0.3520	1.0320	0.8514	0.9235	0.0631	0.3251	0.9874	0.9937

Table E.3: **RMSE** on **Biwi-Kinect** dataset (Fanelli et al., 2013). Best non-Oracle SF results are **bolded**, best non-Oracle results are underlined.

Method	Type	Female → Male				Male → Female			
		Pitch	Roll	Yaw	Mean	Pitch	Roll	Yaw	Mean
DANN (Ganin et al., 2016)	UDA	6.2469	6.7258	4.4961	5.8229	10.0889	7.7898	8.6000	8.8262
RSD (Chen et al., 2021)	UDA	6.2309	6.6824	4.4833	5.7989	10.4134	7.6194	8.2568	8.7632
DARE-GRAM (Nejjar et al., 2023)	UDA	6.3410	6.8664	4.5956	5.9343	10.0607	7.7526	7.6536	8.4890
SSA (Adachi et al., 2024)	DF	6.7116	7.0266	4.6490	6.1291	10.1474	8.0121	8.5636	8.9077
TASFAR (He et al., 2024)	DF	6.8092	7.3885	4.7654	6.3210	12.5730	8.5134	8.8511	9.9792
Source	SF	6.7130	7.1649	4.6409	6.1729	12.2213	8.4473	8.7855	9.8180
BN-adapt (Benz et al., 2021)	SF	6.7960	7.1436	4.6135	6.1843	11.5047	8.1348	8.3536	9.3310
VM	SF	6.5037	7.0540	4.4677	6.0085	11.9837	8.1452	8.5477	9.5589
AugSelfTr	SF	6.6517	7.0615	4.4441	6.0524	11.4350	7.9532	7.8298	9.0726
MERCI	SF	6.2487	7.0180	4.4659	5.9108	10.9115	7.9664	7.0137	8.6306
MERCI w. FN	SF	6.2111	6.9568	4.4298	5.8659	**10.6790**	7.8997	6.5044	8.3610
MERCI-R	SF	**6.1018**	**6.9321**	**4.3859**	**5.8066**	11.1503	7.9469	7.1924	8.7632
MERCI-R w. FN	SF	6.1992	6.9554	4.3939	5.8495	10.7193	**7.8149**	**6.2131**	**8.2491**
Oracle	-	1.1479	1.4720	1.1462	1.2554	1.6862	1.9784	1.5135	1.7260

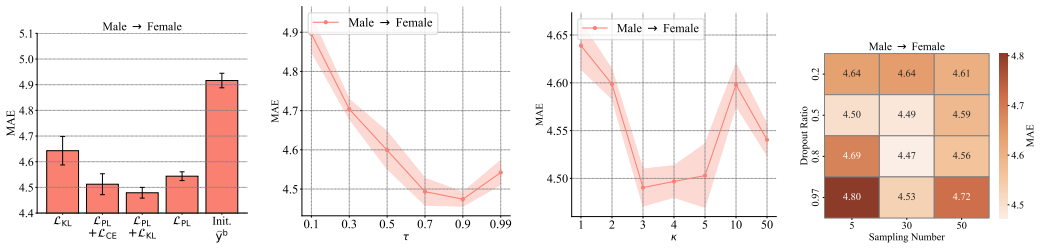
Table E.4: **R** on **Biwi-Kinect** dataset (Fanelli et al., 2013). Best non-Oracle SF results are **bolded**, best non-Oracle results are underlined.

Method	Type	Female → Male				Male → Female			
		Pitch	Roll	Yaw	Mean	Pitch	Roll	Yaw	Mean
DANN (Ganin et al., 2016)	UDA	0.9642	0.7939	0.9882	0.9154	0.9445	0.8236	0.9796	0.9159
RSD (Chen et al., 2021)	UDA	0.9641	0.7948	0.9884	0.9158	0.9368	0.8333	0.9803	0.9168
DARE-GRAM (Nejjar et al., 2023)	UDA	0.9624	0.7730	0.9879	0.9078	0.9420	0.8121	0.9782	0.9108
SSA (Adachi et al., 2024)	DF	0.9612	0.7908	0.9879	0.9133	0.9529	0.8201	0.9826	0.9185
TASFAR (He et al., 2024)	DF	0.9601	0.7390	0.9850	0.8947	0.9277	0.7857	0.9600	0.8911
Source	SF	0.9608	0.7658	0.9878	0.9048	0.9229	0.7906	0.9789	0.8975
BN-adapt (Benz et al., 2021)	SF	0.9615	0.7792	0.9876	0.9094	0.9304	0.8139	0.9809	0.9084
VM	SF	0.9634	**0.7728**	0.9884	0.9082	0.9240	0.8101	0.9791	0.9044
AugSelfTr	SF	0.9621	0.7651	0.9888	0.9053	0.9264	0.8178	0.9788	0.9077
MERCI	SF	**0.9646**	0.7639	0.9886	0.9057	0.9273	0.8181	0.9804	0.9086
MERCI w. FN	SF	0.9640	0.7723	0.9886	**0.9083**	**0.9319**	0.8255	0.9814	0.9129
MERCI-R	SF	0.9640	0.7695	0.9890	0.9075	0.9283	0.8189	0.9814	0.9095
MERCI-R w. FN	SF	0.9645	0.7675	**0.9892**	0.9071	0.9316	**0.8288**	**0.9830**	**0.9144**
Oracle	-	0.9988	0.9911	0.9993	0.9964	0.9982	0.9888	0.9989	0.9953

E.2 Additional Ablation Results and Discussions

Ablation Results. Similar to the ablation study of the histogram head presented in the main paper, we conduct experiments on the Biwi-Kinect dataset under the Male → Female, as shown in Figure E.3a.

Discussion about $\mathcal{L}_{\text{FN}}$. Inspired by prior work (Chen et al., 2021), we note that feature scale is important in regression tasks and may affect adaptation performance. In the Source-Free Domain Adaptive Regression setting, the source model already yields informative predictions on the target domain. To continue benefiting from the source model while ensuring stable and robust adaptation, we introduce the FN term to regularize the feature scale. In practice, we find that while the FN



(a) Ablation Study on f_{hist} Training (b) Confidence Mass Threshold τ (c) Variance Scaling Factor κ (d) Dropout Configs

Figure E.3: *Ablation and hyperparameter sensitivity analysis on the Biwi-Kinect dataset (Male → Female, Yaw)*

Table E.5: Ablation Study on **Moving Average Coefficient (α)** Sensitivity (**UTKFace**).

Moving Average Coef. α	0.01	0.1	0.3	0.5	0.7	0.9	0.99
SOURCE (MAE), female→male	6.5494	6.5494	6.5494	6.5494	6.5494	6.5494	6.5494
MERCI-R (MAE), female→male	6.1334	6.0969	5.9316	5.8263	5.8590	5.9473	6.0270
SOURCE (MAE), male→female	7.9390	7.9390	7.9390	7.9390	7.9390	7.9390	7.9390
MERCI-R (MAE), male→female	7.2304	7.1785	7.1577	7.0802	7.0740	7.0157	7.0780

loss helps maintain a consistent feature scale, it can also restrict the extent of updates to the target model, potentially limiting performance. In cases where the target model can substantially outperform the source model after adaptation (e.g., UTKFace), the gain from FN is marginal. In contrast, for more challenging adaptation tasks (e.g., California Housing), the FN loss offers more noticeable improvements, as shown in Table 2 in the main paper. Additionally, our overall framework, MERCI, already improves adaptation robustness, which reduces the relative importance and observable benefit of the FN term in some scenarios.

E.3 Addiional Hyperparameters Sensitivity Analysis

τ, κ and Dropout Configuration. Similar to the hyperparameter sensitivity analysis presented in the main paper, we conduct experiments on the Biwi-Kinect dataset under the Male → Female, as shown in Figure E.3b- E.3d.

Moving Average Coefficient α. Following previous work (Qiu et al., 2021), we adopt a moving average strategy to update the empirical discrete distribution, $\widetilde{p}_i^{\text{old}}$, in the Histogram Information Set, which stabilizes the adaptation process and preserves useful information from the source model predictions.

In our implementation, we fix the update ratio to 0.5. To further validate the effect of the moving average strategy on histogram head training and the resulting discrete distribution, we conduct additional ablation studies, as shown in Table E.5. When α is small, the histogram used to generate pseudo-labels for the regressor is primarily influenced by the current epoch's output. As α increases, the histogram incorporates more information from the original partial-label set and the prior. Within a broad range of values (0.3–0.9), the experimental results demonstrate stable performance, confirming the both the practicality and robustness of the moving average mechanism.

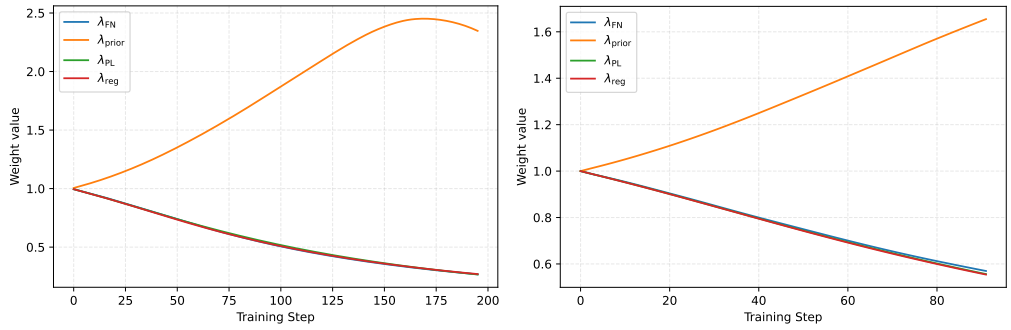
Loss Coefficients. In our experiments, the weighting coefficients are treated as learnable parameters. This design choice aligns the source-free domain adaptation setting, where tuning or selecting hyperparameters manually is often impractical in real-world applications.

To avoid the trivial solution such that $\lambda \to 0$, we follow the well-established uncertainty-based multi-task weighting formulation (Kendall et al., 2018) and take the following steps. First, each weight is parameterized as $\lambda = \exp(-\log \sigma^2)$, and the total loss contains the regularization term $\frac{1}{2}\sum_i \log \sigma_i^2$, which increases when λ becomes too small, thereby preventing collapse. Second, we clamp $\log \sigma_i^2 \in [-10, 10]$ to keep all weights within a stable, non-zero range. Third, NaN/Inf values revert to uniform weighting for numerical safety. The learning weight is reset at the beginning of

Table E.6: Optimized $\boldsymbol{\lambda}$ **Values** for Each Dataset and Shift Setting.

Dataset	Source → Target	λ_{reg}	λ_{PL}	λ_{prior}	λ_{FN}
UTKface	female → male	0.26	0.27	2.31	0.26
UTKface	male → female	0.30	0.30	1.95	0.32
Biwi Kinect	female → male (pitch)	0.37	0.36	1.15	0.38
Biwi Kinect	female → male (yaw)	0.36	0.36	1.74	0.36
Biwi Kinect	female → male (roll)	0.37	0.36	1.23	0.40
Biwi Kinect	male → female (pitch)	0.55	0.56	1.53	0.58
Biwi Kinect	male → female (yaw)	0.56	0.56	1.64	0.56
Biwi Kinect	male → female (roll)	0.56	0.56	1.78	0.75
Digits	SVHN → MNIST	1.09	0.35	2.87	2.99
Digits	MNIST → SVHN	2.26	0.08	1.31	1.45
California House	Near Bay → Far Bay	1.00	1.00	1.00	1.00
California House	Far Bay → Near Bay	1.00	1.00	1.00	1.00

each epoch, and the training dynamics on the UTKFace and Biwi-Kinect datasets are illustrated in Figure E.4.



(a) UTKFace (Female→Male) (b) Biwi-Kinect (Pitch, Male→Female)

Figure E.4: Training trajectories of the learned loss weights on UTKFace and Biwi-Kinect. In both datasets, the weights evolve smoothly and remain within stable, non-zero ranges, showing that our uncertainty-based formulation and regularization prevent the trivial collapse $\lambda \to 0$.

To provide further guidance, we report the final optimized values in Table E.6. Notably, while most weights remain similar, the λ_{prior} values vary across datasets. To assess the sensitivity of performance to this variation, we conducted additional ablation studies on the UTKFace dataset. As shown in Table E.7, our method remains robust to different λ_{prior} values.

Table E.7: Ablation Study (**MAE**) on $\boldsymbol{\lambda}_{\text{prior}}$ Sensitivity (**UTKFace**).

Setting (λ_{reg}-λ_{PL}-λ_{prior}-λ_{FN})	UTKface (female→male)	UTKface (male→female)
0.3-0.3-0.3-0.3	6.00	7.09
0.3-0.3-0.8-0.3	5.95	7.19
0.3-0.3-1.0-0.3	5.99	7.11
0.3-0.3-2.0-0.3	5.96	7.05
0.3-0.3-3.0-0.3	6.01	7.16
0.3-0.3-4.0-0.3	5.95	7.27
1.0-1.0-1.0-1.0	5.99	7.02
Learnable	5.86	7.18

Table E.8: Ablation Study (**MAE**) on **Bin Number** (**UTKFace**).

Bin Number	5	25	45	65	85	100	Self-Adaptive
UTKFace (female → male)	6.55	5.97	5.82	5.88	5.83	5.84	5.83
UTKFace (male → female)	7.94	7.05	7.06	7.08	7.33	7.00	7.08

Bin Size. In the SFDA setting, selecting an optimal bin size is also challenging due to the absence of labeled target data. To address this, we propose a self-adaptive discretization strategy that automatically estimates the number of bins, with results shown in Table 3 in the main paper.

To provide further insight, we conduct an ablation study on various bin numbers using the UTKFace dataset (Table E.8). The results show that model performance remains generally stable across a broad range of bin sizes, but degrades with extremely small bin counts, where coarse discretization renders the histogram labels less informative.

E.4 Robustness of MERCI under Different Levels of Distribution Shift

Table E.9: Performance of MERCI-R under **Shifted** UTKFace (**MAE**).

Severity	SOURCE (Female → Male)	MERCI-R (Female → Male)	SOURCE (Male → Female)	MERCI-R (Male → Female)
N/A	6.549	5.826	7.939	7.080
Light	12.635	9.081	11.374	9.995
Mild	15.667	11.837	15.261	12.137
Severe	19.067	15.554	21.927	14.775
Extremely Severe	22.082	21.396	23.686	20.302

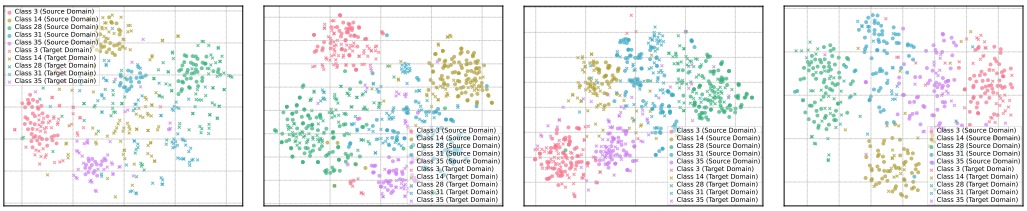
Table E.10: Performance of MERCI-R on Toy Data with **Shift Severity** (**MAE**).

Shift Severity	0.1	0.15	0.2	0.25	0.375	0.5	1	1.5
MERCI-R (MAE)	0.168	0.304	0.625	0.882	1.265	1.716	3.279	4.704
Source (MAE)	0.318	0.518	0.813	1.008	1.482	1.926	3.530	4.927
Avg. Partial Label Set Size	4.083	4.101	4.133	4.190	4.285	4.670	6.102	7.565

To evaluate MERCI's performance under substantial distribution shifts, we consider two settings: (1) A real-world dataset with Gaussian noise, where larger mean values induce stronger shifts; (2) A 2D toy dataset with polynomial-based inputs and controllable linear shifts.

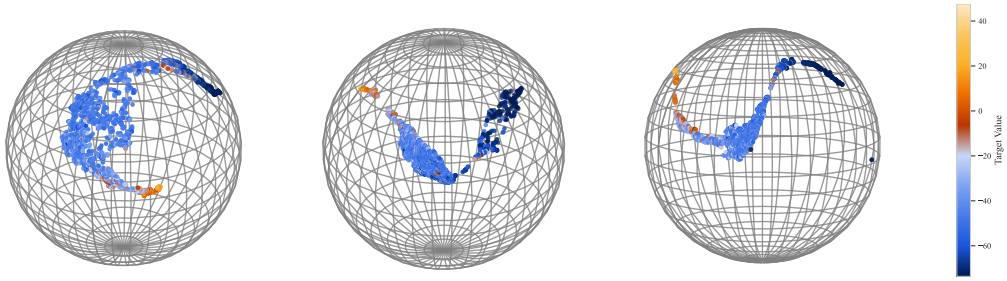
The corresponding results are reported in Table E.9 (UTKFace) and Table E.10 (Toy Dataset). For Table E.9, we introduce Gaussian noise into the target domain at varying severity levels using the public corruption library *imagenet-c*. This library controls noise via a severity parameter ranging from 0 to 5. A severity of "N/A" indicates no added noise, so only natural domain shift is present; severity 1 corresponds to a "light" setting, severity 2 to "mild", severity 3 to "severe", and severity 5 to "extremely severe". Each severity level maps to a predefined standard deviation of Gaussian noise (e.g., 0.08 for severity 1 and 0.38 for severity 5), allowing us to simulate progressively stronger domain shifts in a controlled manner. As for Table E.10, shift severity is quantified by the central distance between source and target inputs.

The proposed MERCI framework demonstrates overall robustness across different shift magnitudes and performs well under slight to moderate shifts. In extreme shift scenarios, MERCI's reliance on the source model's predictions for target data limits its performance. Specifically, it struggles to generate reliable estimates of $\hat{y}^b$ when the domain gap is too large. This reflects a realistic constraint: in the absence of source data, adaptation becomes fundamentally challenging when the target domain diverges significantly from the source. Nevertheless, MERCI provides uncertainty estimates for pseudo-labels—for instance, the average length of partial label sets (as shown in Table E.10), which can serve as a proxy indicator of shift severity.



(a) Ar to Cl (b) Cl to Pr (c) Pr to Rw (d) Rw to Ar

Figure E.5: Illustration of cluster feature representations in the SFDA classification task on the Office-Home dataset (Venkateswara et al., 2017). The plots compare source and target feature distributions using t-SNE (Van der Maaten & Hinton, 2008) on a general object classification task. In the figures, points represent source-domain data and crosses represent target-domain data, while different colors indicate different classes. Both source and target inputs are processed through the pre-trained source model, which also serves as the initialization for the target model. Dots represent source data, crosses represent target data, and different colors correspond to five randomly selected classes. In the source domain, clear clusters and well-separated class boundaries are observed. In contrast, target samples show clustering tendencies but are more diversely distributed.



(a) Target Data from Source Model (b) Target Data after Adaptation (Without f_{hist}) (c) Target Data after Adaptation (MERCI-R)

Figure E.6: *UMAP visualization of target data feature space on UTKFace dataset*

E.5 Cluster Property in Feature Space of Classification Problem

In classification, discrete labels naturally guide deep models to learn clustered feature embeddings, a phenomenon aligned with the cluster assumption in classification and semi-supervised learning (Chapelle et al., 2009). In our work, the term "clustered feature space" in the Sec. 1 exactly refers to this phenomenon. More specifically, feature representations learned by deep models in classification tasks are separable by class, i.e., samples from the same class form clusters in embedding space that are separated by low-density boundaries. This phenomenon has been observed across various classification tasks and data modalities, including text topic classification (Lee et al., 2024), speech emotion classification (Kim et al., 2017), and animal image classification (Yang et al., 2022a). It has also been widely observed in SFDA classification benchmarks (Liang et al., 2020; Yang et al., 2021).

In contrast, regression involves continuous targets, encouraging smooth feature-to-output mappings without explicit class separation. As a result, deep regression features typically lie on a continuous manifold and lack the clustering structure seen in classification (Islam et al., 2023).

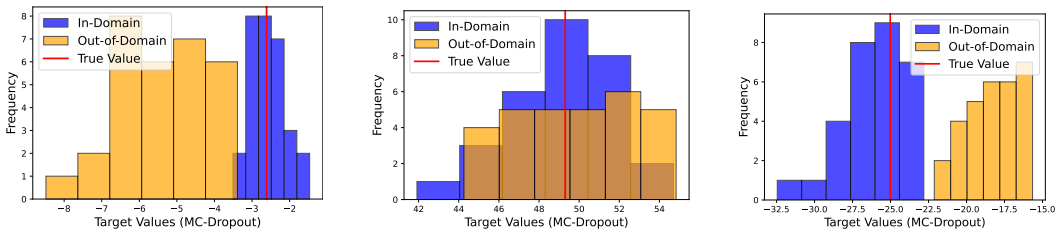
Our observation of clustered structures in the feature space for classification emerged from empirical experiments (see Figure E.5), and was used to highlight the different feature manifolds observed in regression (see Figures 4 and E.6), which may not form clearly separable clusters and thus introduce unique challenges for pseudo-label refinement.

E.6 Feature Representation Learning in Deep Regression Models

Due to the page limitation of the main paper, we provide additional observation on feature representations learned by deep regression model in this Appendix.

In Figures 4 and E.6, we visualize three types of target domain feature representations on both UTKFace (Male $\rightarrow$ Female) and Biwi-Kinect (Female $\rightarrow$ Male, pitch angle prediction) datasets : **(a)** direct feature representations of target data passed through the pre-trained source model; **(b)** adapted target feature obtained by training with only the conventional regression loss ($\mathcal{L}_{\mathrm{RMSE}}$), using initially collected histogram information as supervision signals; **(c)** adapted target feature under our complete MERCI-R framework.

Comparing (a) with (b) and (c), the adapted features exhibit more locally compact structures, reflecting the controlled conditional entropy $\mathcal{H}(\mathrm{Z}|\mathrm{Y})$. However, a closer comparison between (b) and (c) shows that the complete MERCI-R framework, equipped with the histogram head and classification loss, achieves both local compactness and global diversity. By contrast, relying solely on MSE or RMSE loss tend to reduce feature entropy $\mathcal{H}(\mathrm{Z})$, which limits generalizability and degrades performance. When the target values are noisy, as in SFDAR settings, the performance of MSE loss becomes unstable, hindering smooth representation learning (Figure 4(b)) and showing weakness in handling extreme values (Figure E.6(b)).



(a) Example 0 (b) Example 1 (c) Example 2

(d) Example 4 (e) Example 5 (f) Example 6

Figure E.7: MC-Dropout predictive distributions for samples from the female domain on Biwi-Kinect dataset. In-domain predictions (blue) exhibit clear unimodality, while out-of-domain predictions (yellow) deviate from the mean and lack unimodality. These results empirically motivate the use of a Gaussian prior for the histogram head.

E.7 Unimodal Conditional Distribution Validation

In this subsection, we explain the unimodality assumption used in MERCI, supported by both theoretical insights and experimental observations.

Unimodal vs. Multimodal. When the input X contains sufficient information to predict Y, the uncertainty around the prediction is commonly assumed to be dominated by measurement or annotation noise, such as Gaussian or sub-Gaussian behavior. In this setting, the conditional predictive distribution tends to concentrate within a single continuous interval, rather than exhibiting genuinely multimodal structure. Multimodality typically appears when essential information is missing. For example, predicting age from height alone without sex information may lead to two separated plausible ranges. In contrast, given a clear frontal facial image, we intuitively expect age to fall within a contiguous region with high probability instead of forming two distinct clusters, such as around 25 or 45. The target tasks in our experiments, including age estimation on UTKFace and head pose estimation, provide rich visual cues, so the unimodal assumption is a reasonable modeling choice. We also note that our framework can accommodate multimodal scenarios. Specifically, we can (1) model each peak with a separate unimodal component, or (2) incorporate richer prior information describing the multimodal structure and apply it on top of the partial-label set.

Gaussian prior. In our experiments, we chose a Gaussian prior for two practical reasons. First, the Gaussian distribution is a standard choice that allows closed-form computation of the likelihood and KL term. Second, when only partial knowledge of a distribution is available (in SFDA setup), selecting a Gaussian prior follows the maximum-entropy principle, meaning that the distribution that best reflects the current state of knowledge is the one with the largest entropy. In particular, the Gaussian maximizes differential entropy under a fixed variance constraint, so it introduces no additional structural assumptions beyond the observed scale of label variability (Rioul, 2021). In our method, dropout sampling together with histogram estimation provides information on the mean and approximate dispersion of the label distribution, making the Gaussian a suitable default in the absence of further empirical evidence. More complex priors are only warranted when such empirical evidence is present. Specifically, if additional structural information about the label distribution becomes available, other priors can be naturally incorporated into our framework. For instance, a Laplace prior may be appropriate when residuals are known to exhibit sharp, heavy-tailed behavior, whereas Gaussian mixtures or non-parametric priors are suitable when multimodality is empirically observed.

Empirical Evidence. To examine the unimodality assumption of the conditional distribution $P_{\mathrm{y}|\mathrm{x}}$, we randomly select several samples from the female domain of the Biwi-Kinect dataset and evaluate them using two models: (1) an in-domain model trained on female-domain data, and (2) an out-of-domain model trained on the male domain. We estimate and visualize the MC-Dropout distributions for each sample to enable a direct comparison and more detailed analysis. As shown in Figure E.7, the MC-Dropout outputs of the well-trained in-domain model exhibit a clear unimodal pattern. In contrast, the out-of-domain model often produces distributions that deviate substantially from the mean, lack unimodality, and display pronounced skewness. These observations empirically motivate the adoption of a Gaussian prior when training the histogram head.

E.8 Analysis of Mutual Enhancement Between Histogram and Regression Heads

To examine the directional contributions within MERCI, such a mutual-enhancement framework, we conducted an additional set of experiments on the UTKFace dataset. In particular, we selectively disabled two forms of feedback: (1) the feedback from the histogram head to the regression head after the warm-up stage, and (2) the feedback from the regression head to the histogram head after initialization. The resulting MAE values are presented in Table E.11.

As shown in the table, removing the histogram-to-regression feedback leads to a noticeably larger performance degradation, suggesting that this direction plays a more critical role in improving prediction quality. In contrast, removing the regression-to-histogram feedback results in a smaller degradation; however, the initialization of the histogram head still fundamentally relies on the regressor. Therefore, this pathway as well as the overall design of the framework remains unified and mutually consistent.

Table E.11: MAE evaluation for directional ablations within the mutual-enhancement framework.

Model	Female → Male	Male → Female
Full Model	**5.82**	**7.0802**
regression $\not\rightarrow$ histogram	6.0492	7.1080
histogram $\not\rightarrow$ regression	6.1304	7.1704

Table E.12: Efficiency comparison of MERCI and baseline methods.

Method	Init Time (s)	Train Time / Epoch (s)	Peak GPU Memory
SSA (Adachi et al., 2024)	86.97	190.31	5.33 GB
BN-adapt (Benz et al., 2021)	0.00	170.91	0.73 GB
RSD (Chen et al., 2021)	0.00	202.96	10.66 GB
DANN (Ganin et al., 2016)	0.00	194.63	10.76 GB
DARE-GRAM (Nejjar et al., 2023)	0.00	308.46	10.97 GB
VM	0.00	240.49	21.04 GB
TASFAR (He et al., 2024)	435.71	180.28	5.51 GB
MERCI-R	305.32	232.94	6.41 GB

Table E.13: Parameter count of MERCI components.

Component	Parameters
Backbone + Regressor	120.30M
Histogram Head	0.53M

E.9 Efficiency and Scalability Analysis

In this section, we provide a detailed efficiency and scalability analysis of our proposed method, MERCI, including (i) initialization time, (ii) per-epoch training time, (iii) peak GPU memory usage, and (iv) parameter count on Biwi-Kinect dataset. All experiments were conducted on a single NVIDIA GPU under the same settings as the main paper for a fair comparison.

Initialization Time. MERCI includes a one-time initialization step that involves stochastic forward passes and a density estimation. As shown in Table E.12, the initialization takes about 305 seconds, of which fewer than 10 seconds correspond to KDE estimation. The total initialization cost is comparable to that of TASFAR (around 430 seconds).

Per-Epoch Training Time. MERCI's per-epoch training time (about 230s) is on the same order as other source-free and unsupervised domain adaptation baselines, such as SSA (190.31 s), RSD (202.96 s), DANN (194.63 s), DARE-GRAM (308.46 s), VM (240.49 s), and TASFAR (180.28 s). Despite including an initialization stage, the iterative training cost of MERCI remains competitive.

Peak GPU Memory Usage We also report peak GPU memory usage in Table E.12. MERCI requires only 6.41 GB, which is substantially lower than traditional unsupervised domain adaptative regression methods such as RSD, DANN, DARE-GRAM, and SFDAR method VM. This demonstrates that MERCI is computationally manageable and memory-efficient, enabling scalability to common backbone architectures.

Parameter Count. Finally, Table E.13 summarizes the parameter count. The backbone plus regressor contains about 120.30M parameters, while the proposed histogram head adds only 0.53M (<0.5% overhead), indicating that MERCI introduces minimal and controllable architectural complexity.

E.10 Calibration Analysis of Source Model on Target Domain

To evaluate how well the uncertainty from the source model transfers to the target domain, we conduct a calibration analysis using two stochastic prediction mechanisms: MC-Dropout (MC-D) and Augmentation Ensemble (Aug-Ens). For each target sample, multiple stochastic predictions are generated. Under a Gaussian approximation of these sampled predictions, we estimate a predictive mean μ_i and standard deviation σ_i, and construct a predictive interval $\hat{I}_i$ at confidence level P as:

$$\hat{I}_i = \left[\, \mu_i - Z_P \sigma_i,\ \mu_i + Z_P \sigma_i \,\right],$$

where Z_P is the standard normal quantile corresponding to confidence level P. This Gaussian-based interval approximates the range that the model expects to contain the true value with probability P.

To quantify empirical calibration, we use the Prediction Interval Coverage Probability (PICP), defined as the fraction of target samples whose ground-truth values fall within the predictive interval $\hat{I}_i$:

$$\text{PICP} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}[\, y_i \in \hat{I}_i].$$

A well-calibrated estimator should satisfy PICP $\approx P$.

The results in Table E.14 show that MC-D is consistently under-calibrated on the target domain, and Aug-Ens exhibits even stronger under-coverage across all confidence levels and variables. This indicates that the miscalibration arises not from a specific estimator but from the inherent distribution shift between source and target domains. These observations also clarify why uncertainty-dependent SFDA methods such as TASFAR perform poorly, and they motivate MERCI's use of partial label sets and a histogram head to mitigate miscalibrated uncertainty and better capture the target-domain density structure.

Table E.14: Calibration of MC-Dropout (MC-D) vs. Augmentation Ensemble (Aug-Ens) on Biwi-Kinect (female → male). Target confidence P denotes the nominal CI level; PICP is the empirical coverage.

Variable	Target P	MC-D PICP	MC-D $\lvert\text{PICP} - P\rvert$	Aug-Ens PICP	Aug-Ens $\lvert\text{PICP} - P\rvert$
pitch	50%	0.3006	0.1994	0.2546	0.2454
pitch	60%	0.3662	0.2338	0.3147	0.2853
pitch	70%	0.4390	0.2610	0.3822	0.3178
pitch	80%	0.5211	0.2789	0.4693	0.3307
pitch	90%	0.6405	0.2595	0.5793	0.3207
roll	50%	0.1938	0.3062	0.1643	0.3357
roll	60%	0.2461	0.3539	0.2070	0.3930
roll	70%	0.2978	0.4022	0.2622	0.4378
roll	80%	0.3657	0.4343	0.3231	0.4769
roll	90%	0.4721	0.4279	0.4273	0.4727
yaw	50%	0.1790	0.3210	0.1382	0.3618
yaw	60%	0.2296	0.3704	0.1711	0.4289
yaw	70%	0.2880	0.4120	0.2078	0.4922
yaw	80%	0.3572	0.4428	0.2545	0.5455
yaw	90%	0.4562	0.4438	0.3188	0.5812