SHORTCUT TO NOWHERE: DEMYSTIFYING DEEP SPURIOUS REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world regression often exhibits shortcuts: attributes that are spuriously correlated with continuous targets in training, yet unreliable under deployment shifts; regressing targets using such shortcuts may fail catastrophically at test time. Existing studies on spurious correlations focus primarily on classification, where labels are categorical and groups are naturally defined. However, many real-world tasks require continuous prediction, where hard label boundaries or discrete group-label pairs do not exist. We define Deep Spurious Regression (DSR) as learning from regression data with attribute-label confounding, addressing continuous spurious correlations, and generalizing to all attribute-label combinations at test time. Motivated by the intrinsic difference between classification and regression shortcuts, we propose to exploit the similarity among spurious attributes in both label and feature spaces, thereby accounting for nearby targets and related groups while calibrating both label and learned feature distributions across attributes. Extensive experiments on common real-world DSR datasets that span computer vision, environmental sensing, and large language model (LLM) regression verify the superior performance of our strategies. Our work fills the gap in benchmarks and techniques for studying spurious correlations in continuous prediction.

1 INTRODUCTION

Spurious correlations are ubiquitous and inherent in real-world observational data (Yang et al., 2024; 2023b). Rather than preserving a stable relationship between target-relevant features and labels, the data often exhibit shortcut correlations, where certain attributes are highly predictive of the target during training but unreliable at deployment (Geirhos et al., 2020). This phenomenon poses great challenges for deep learning models and has motivated many prior techniques for addressing spurious correlations and subgroup failures (Sagawa et al., 2020; Liu et al., 2021; Nam et al., 2020; Kirichenko et al., 2023; Creager et al., 2021; Zhang et al., 2022; Holste et al., 2024).

Existing solutions for spurious correlations, however, focus on targets with ***categorical*** indices, i.e., the targets are different classes, and subgroups can be naturally defined by finite label-attribute combinations. However, many real-world tasks involve ***continuous*** and even infinite target values, where discrete group definitions do not exist. Unfortunately, standard models trained on such data learn fragmented and shortcut-driven mappings that are incapable of capturing the continuous relationships that underlie regression tasks. Fig. 1 illustrates this failure mode on `ColoredRotatedMNIST`, where the target is the rotation angle and the spurious attribute is the background color (details in Appendix C.1). Rather than learning the continuous rotation structure, standard empirical risk minimization (ERM) produces error patterns tied to the *color-angle shortcut*: some low-data regions have low error, while others fail sharply when the shortcut breaks. Such a *continuous* and *attribute-dependent* error distribution is suboptimal for regression and cannot be fully captured by discrete group counts.

In this work, we systematically investigate Deep Spurious Regression (DSR) arising in real-world settings. We define DSR as learning continuous targets from data with attribute-label confounding, dealing with potentially sparse or missing data for certain attribute-label combinations, and generalizing to a test set that is balanced over the entire range of continuous target values and spurious attributes. This definition is analogous to the spurious correlation problem in classification (Sagawa et al., 2020), but focuses on the continuous setting.

DSR brings new challenges distinct from its classification counterpart. First, given continuous and potentially infinite target values, discrete groups are no longer naturally defined, causing ambiguity

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

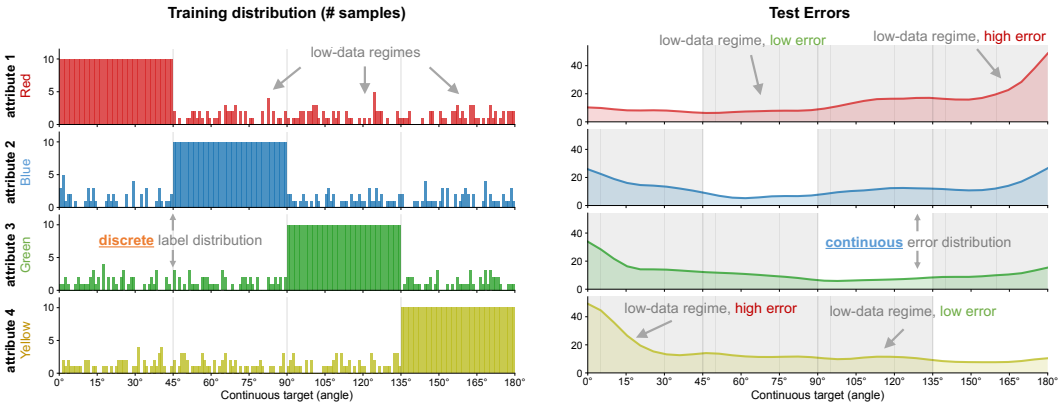

Figure 1: **Example illustration of Deep Spurious Regression (DSR). Left:** In `ColoredRotatedMNIST`, each spurious attribute, represented by a background color, is strongly associated with a dominant angle range in training, while other angle ranges have few samples. **Right:** ERM produces continuous test-error curves across the target angle. Importantly, low-data regimes do not always lead to high error: certain sparse regions remain easy, while others fail sharply when the color-angle shortcut breaks. This shows that DSR cannot be captured by discrete group counts alone. More details are in Appendix C.1.

when directly applying traditional debiasing methods such as re-sampling, re-weighting, and group-robust optimization (Sagawa et al., 2020). Second, both nearby labels and related attributes carry meaningful information for interpreting spurious correlations. For example, two weakly observed targets under one attribute may differ substantially if one is supported by nearby labels or similar attributes, while the other lies in a sparse neighborhood. Finally, unlike classification, certain attribute-label combinations may have no data at all, which motivates the need for interpolation and extrapolation across targets and attributes.

To fill these gaps, we propose two simple yet effective methods for addressing DSR: label multi-dimensional scaling (`L-MDS`) and feature multi-dimensional scaling (`F-MDS`). A key idea underlying both approaches is to leverage the *similarity* among spurious attributes by pooling information across similar groups and attributes using Multi-Dimensional Scaling (MDS) (Borg & Groenen, 2005) and kernel smoothing to perform explicit calibration in the label and feature spaces. Both techniques can be easily embedded into existing deep networks or large language models (LLMs) and allow optimization in an end-to-end fashion. We verify that our techniques not only calibrate for the intrinsic underlying spurious structure, but also provide large and consistent gains when combined with standard regression objectives.

To support practical evaluation of spurious regression, we curate and benchmark DSR datasets for common real-world tasks in computer vision, environmental sensing, and natural language processing. They range from visual regression tasks such as age prediction, to LLM regression tasks such as code metric prediction. We further set up benchmarks for proper DSR performance evaluation. Our contributions are as follows:

- We formally define DSR as regression learning with spurious correlations, generalizing to the entire target range and all attribute-label combinations. DSR provides thorough and unbiased evaluation of learning algorithms in practical continuous prediction settings.
- We develop two simple, effective, and interpretable algorithms for DSR, `L-MDS` and `F-MDS`, which exploit the similarity among spurious attributes in both label and feature spaces.
- We curate benchmark DSR datasets in different domains: computer vision, environmental sensing, and natural language processing, covering diverse tasks from synthetic data to LLM regression. We set up strong baselines as well as benchmarks for proper DSR performance evaluation.
- Extensive experiments on real-world DSR datasets verify the superior performance of our strategies. We further reveal intriguing properties of DSR on robustness and generalization.

2 Related Work

Spurious Correlations in Classification. Spurious correlations, which arise when a model relies on a feature correlated with the target during training but not causally related to it, have been widely

documented in classification tasks (Geirhos et al., 2020). This phenomenon, also termed *shortcut learning*, is closely tied to the tendency of neural networks to exploit the simplest available signal (Shah et al., 2020). For example, classifiers trained on biased datasets may rely on background cues (Beery et al., 2018), textures (Geirhos et al., 2019), or demographic attributes (Buolamwini & Gebru, 2018; Sagawa et al., 2020), rather than semantically meaningful features. Benchmarks such as SubpopBench further systematize this challenge with shifts across diverse domains (Yang et al., 2023b). However, existing works have focused exclusively on classification, where labels are discrete and groups can be naturally defined by label-attribute pairs. In contrast, we study the underexplored *regression* setting, where the target is real-valued and such discrete group no longer directly applies.

Mitigating Spurious Correlation in Classification. Existing mitigation methods for spurious correlations broadly aim to reduce a model's reliance on non-causal but predictive shortcuts. One line of work uses distributionally robust optimization to emphasize high-loss or underperforming groups, often assuming that group annotations are available during training (Duchi & Namkoong, 2021; Sagawa et al., 2020). Another line identifies biased or misclassified samples with a preliminary model, and then upweights them to improve worst-group performance (Liu et al., 2021; Nam et al., 2020). Other approaches encourage invariant representations across environments, suppress domain-discriminative information, or retrain the classifier on a more balanced subset to reduce shortcut reliance (Ganin et al., 2016; Arjovsky et al., 2019; Kirichenko et al., 2023; Zhang et al., 2022). Despite this progress, these methods largely assume discrete labels or finite group-label pairs, and therefore do not directly generalize to regression settings where targets are continuous and label ordering carries semantic meaning.

Regression Learning and Imbalanced Regression. Real-world prediction problems often require estimating *continuous* targets from observational data, including visual, environmental, language, and physiological signals (Zhang et al., 2017; Hu et al., 2019; Yang et al., 2023a). Deep regression models are typically trained to minimize average prediction error, which can be insufficient when target values are long-tailed or shifted across domains (Branco et al., 2016). Recent work on Deep Imbalanced Regression improves learning across the full target range by smoothing label and feature distributions over nearby targets (Yang et al., 2021), while other methods reformulate regression losses, balance cross-domain transfer, or learn representations that preserve the ordinal structure of continuous labels (Ren et al., 2022; Yang et al., 2022; Zha et al., 2023). However, these methods mainly address marginal label imbalance, domain imbalance, or target-aware representation learning, and do not explicitly model *spurious* attribute correlations in regression. Our work fills this gap by modeling geometric relationships among spurious attribute groups, directly addressing the joint (y, a) imbalance and continuous spurious correlations that existing regression methods overlook.

3 METHODS

Problem Setup. We consider supervised regression with training data $\mathcal{D} = \{(\mathbf{x}_i, y_i, a_i)\}_{i=1}^N$, where $\mathbf{x} \in \mathcal{X}$ is the input, $y \in \mathcal{Y} \subset \mathbb{R}$ is a continuous target, and $a \in \mathcal{A}$ is the spurious attribute. The attribute a is correlated with y in the training distribution, but is unreliable at test time, inducing a shift $p_{\text{train}}(y \mid a) \neq p_{\text{test}}(y \mid a)$. A model trained by ERM (Vapnik, 1998) may exploit this attribute-label correlation as a shortcut, encoding a rather than the true causal features of y, and thus fail to generalize uniformly across all (y, a) combinations at test time.

For density estimation and subgroup analysis, we discretize $\mathcal{Y}$ into K non-overlapping bins $\{B_k\}_{k=1}^K$ following (Yang et al., 2021), assigning each sample a bin index $b_i = k$ if $y_i \in B_k$. A *group* is then defined as a unique combination of bin index and attribute value:

$$g = (k, a) \in \{1, \dots, K\} \times \mathcal{A}, \quad n_g = |\{i : b_i = k,\, a_i = a\}|\,. \tag{1}$$

Unlike classification, these groups are not natural task labels, but auxiliary partitions of a continuous target space. Next, we identify two structural properties of DSR that motivate our approach: continuity in the target space $\mathcal{Y}$ and similarity structure in the attribute space $\mathcal{A}$.

Observation 1: Target continuity enables within-attribute smoothing. We first compare classification and regression under the same spurious structure using two MNIST-based tasks (LeCun et al., 1998) with colored backgrounds as spurious attributes. In both tasks, the spurious attribute is the background color $a \in \{\text{Red}, \text{Blue}, \text{Green}\}$, and each color is strongly associated with a dominant target region in the training set. The *classification* task uses digit identity (i.e., digit "1" to "9") as the categorical label, while the *regression* task uses a rotated digit "2" and takes the rotation angle as the

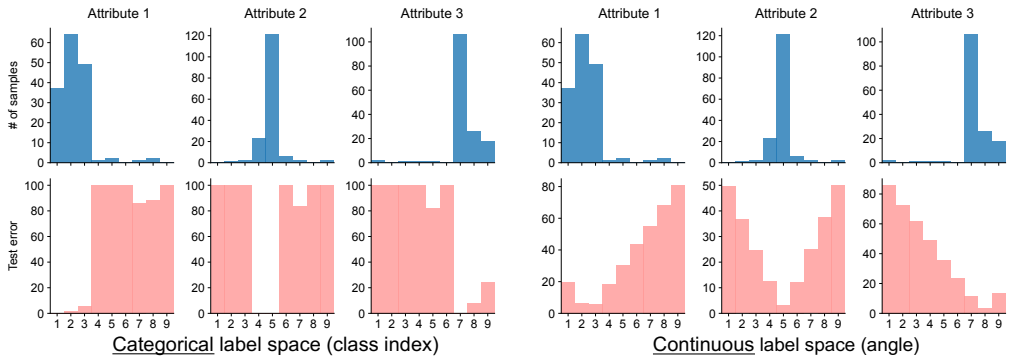

Figure 2: **Classification *vs.* regression under the same spurious structure. Top:** We construct classification and regression tasks with identical per-attribute training distributions, with three spurious attributes and target bins. **Bottom left:** In classification, test error is nearly binary across off-diagonal class-attribute combinations. **Bottom right:** In regression, test error changes smoothly with the distance from the dominant target region.

continuous label. Both tasks have identical per-attribute training distributions and balanced test sets over all target-attribute combinations. As shown in Fig. 2, the classification setting produces nearly binary off-diagonal failures: once a class falls outside the dominant region of an attribute, the model has no notion of how close or far that class is from the training support. In contrast, the regression setting produces graded errors: prediction error changes smoothly as the angle moves away from the dominant target region of each attribute. This reveals a key structure unique to regression: Target bins should not be treated as unrelated classes, because nearby target values carry useful information for each other within the same attribute. We therefore exploit label continuity by smoothing along the target axis (Yang et al., 2021) inside each attribute group.

Observation 2: Attribute similarity enables cross-attribute smoothing. Target continuity alone does not fully address DSR because spurious attributes may have different but related target distributions. To illustrate this, we construct three synthetic distribution scenarios with three attributes $a \in \{\text{Red}, \text{Blue}, \text{Green}\}$. In the ***aligned*** setting, all three attributes have similar target distributions; in the ***partially aligned*** setting, two attributes have similar distributions while the third occupies a different target range; and in the ***misaligned*** setting, all attributes occupy distinct target regions. Fig. 3 shows that the learned feature-space embeddings under ERM reflect these distributional relationships. When attributes share similar target distributions, their per-bin feature centroids are mixed or close to each other (Fig. 3a). When one attribute differs, its embeddings separate from the other two (Fig. 3b). When all attributes are distributionally distinct, the embeddings form separated attribute-specific clusters (Fig. 3c). This suggests that spurious attributes should not be treated as isolated groups. Instead, attributes with similar label or feature distributions can share information, especially for sparse target bins. We therefore smooth not only along the continuous target axis, but also across related attributes through an attribute affinity structure.

Two-Dimensional Distribution Smoothing. Together, the observations above motivate a distribution smoothing strategy that operates along two axes. Along the ***target axis***, we exploit label continuity within each spurious attribute, so that nearby target values can support each other (i.e., training data with nearby target values can borrow statistical strength from each other). Along the ***attribute axis***, we pool information across related attributes, so that attributes with similar target or feature distributions can share statistical strength. The resulting smoothed distributions are then used to derive sample weights for training. Together, these two forms of smoothing address the joint imbalance over (y, a) that is not captured by marginal label reweighting or discrete group-level debiasing alone.

Along the Target Axis. For each attribute $a \in \mathcal{A}$, the target values $y_i : a_i = a$ may follow an uneven distribution across bins. We apply label distribution smoothing (LDS) (Yang et al., 2021) within each attribute, yielding a kernel-smoothed density estimate $\widehat{p}_a(y)$ that captures within-attribute label frequency. Each sample receives a target-axis weight inversely proportional to this density:

$$w_i^{\text{LDS}} = \frac{1}{\widehat{p}_{a_i}(y_i)^{\alpha}},$$

where α controls the reweighting strength. This preserves the idea of LDS, but applies it conditionally on the spurious attribute, instead of estimating a single marginal density over all samples.

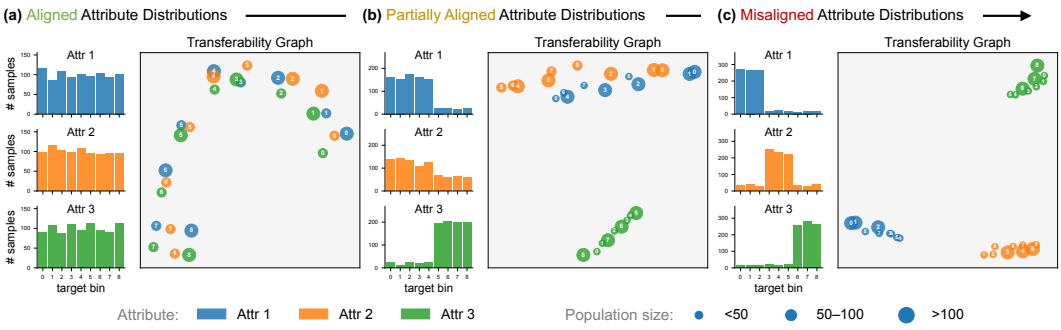

Figure 3: **From aligned to misaligned spurious attributes.** We vary the similarity of target distributions across attributes: **(a)** all attributes share similar target distributions, **(b)** two attributes are similar while one differs, and **(c)** all attributes have distinct target distributions. The learned feature-space embeddings reflect these distributional relationships, motivating MDS-based information sharing across related attributes.

Along the Attribute Axis. Target-axis smoothing alone treats each attribute independently. However, as shown in Fig. 3, attributes with similar target or feature distributions can provide useful information to each other, especially in sparse target regions. We construct an affinity matrix $K \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$, where $K_{aa'}$ measures how much information attribute a' contributes to attribute a. For each group (b, a), let $n_{(b,a')}$ be defined as in Eqn. (1), the smoothed count and per-sample weight are

$$\widetilde{n}_{(b,a)} = \sum_{a'} K_{aa'}\, n_{(b,a')}, \qquad w_i^{\mathrm{MDS}} = \frac{1}{\widetilde{n}^{\alpha}_{(b_i,\,a_i)}}.$$

The final sample weight combines target-axis and attribute-axis smoothing, $w_i = w_i^{\mathrm{LDS}} \cdot w_i^{\mathrm{MDS}}$, and training minimizes the weighted ℓ_1 loss:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} w_i \left\| f_\theta(\mathbf{x}_i) - y_i \right\|.$$

The central question is how to define the affinity matrix K. We propose two instantiations following the same pipeline. Given a pairwise distance matrix $\mathbf{D}$ between attributes, we embed the attributes into a Euclidean space via Multi-Dimensional Scaling (MDS) (Borg & Groenen, 2005), obtaining coordinates $\mathbf{Z} = [z_m]_{m=1}^{|\mathcal{A}|} \in \mathbb{R}^{|\mathcal{A}|\times 2}$ that best preserve the pairwise distances in $\mathbf{D}$. We then apply a row-normalized RBF kernel:

$$K_{mn} = \frac{\exp\!\left(-\|z_m - z_n\|^2/2\tau^2\right)}{\sum_{n'} \exp\!\left(-\|z_m - z_{n'}\|^2/2\tau^2\right)}, \tag{2}$$

τ is the median pairwise distance in $\mathbf{Z}$. The two instantiations differ only in how $\mathbf{D}$ is computed: `L-MDS` uses pairwise Wasserstein distances (Villani, 2009) between per-attribute target distributions, while `F-MDS` uses pairwise Euclidean distances between per-attribute feature centroids.

3.1 Label-MDS: Kernel from Target Distributions

Label-MDS (`L-MDS`) defines attribute similarity through the label distributions. Intuitively, two attributes should be close if their training targets follow similar distributions, as they can provide useful count information to each other during attribute-axis smoothing. This is especially important in DSR, where related attributes may contain nearby target evidence.

Wasserstein distance between attributes. Let $\widehat{p}_a$ be the empirical target distribution of a, estimated from training. We measure pairwise dissimilarity between attributes via Wasserstein-1 distance (Villani, 2009):

$$D_{aa'} = W_1(\widehat{p}_a, \widehat{p}_{a'}), \quad a, a' \in \mathcal{A},$$

forming $\mathbf{D} \in \mathbb{R}^{|\mathcal{A}|\times|\mathcal{A}|}$. We use the Wasserstein-1 distance because it respects the geometry of the continuous space (Peyré & Cuturi, 2019): two distributions supported on nearby targets are considered close even when their supports do not exactly overlap (Rubner et al., 2000). This property is well suited for DSR, where sparse targets can make divergence-based measures such as χ^2 or Kullback–Leibler unstable or ill-defined (Csiszár & Shields, 2004).

Kernel construction. Given $\mathbf{D}$, we apply the shared MDS pipeline: embed the attributes into $\mathbf{Z}$, form the RBF kernel K as in Eqn. (2), and use K for attribute-axis smoothing. Since $\mathbf{D}$ depends only on the training targets, `L-MDS` is computed once before training and adds no additional overhead.

3.2 Feature-MDS: Kernel from Learned Representations

`L-MDS` measures similarity from target distributions alone, which is fixed before training and does not reflect how the model represents different attributes. Feature-MDS (`F-MDS`) instead computes attribute similarity in the learned representation space, yielding a kernel that adapts to the model's evolving feature geometry. This allows attribute-axis smoothing to become progressively aligned with the internal structure learned by the encoder.

Centroid distances in representation space. Let ϕ denote the encoder. At every T epochs, we perform a forward pass over the training set and extract ℓ_2-normalized features $h_i = \phi(\mathbf{x}_i)/\|\phi(\mathbf{x}_i)\|_2$ for all training samples. Normalization makes distances scale-invariant across training stages. For each attribute $a \in \mathcal{A}$, we compute the attribute centroid

$$c_a \;=\; \frac{1}{n_a} \sum\nolimits_{i:\, a_i = a} h_i,$$

The pairwise distance matrix is then defined by Euclidean distances between centroids:

$$D_{aa'} \;=\; \|c_a - c_{a'}\|_2, \quad a, a' \in \mathcal{A}.$$

Kernel construction. Given $\mathbf{D}$, we apply the same MDS pipeline as `L-MDS`: embed the attributes into $\mathbf{Z}$ and form the row-normalized RBF kernel K using Eqn. (2). Attributes that are close in the encoder feature space contribute more to each other's count smoothing, while distant attributes contribute less. As a result, `F-MDS` adaptively transfers information across attributes according to the model's learned representation geometry, rather than relying only on fixed target distributions.

4 Benchmarking DSR

Datasets. To rigorously evaluate spurious regression across a broad range of domains and tasks, we curate and benchmark diverse DSR datasets spanning computer vision, environmental sensing, and natural language processing. Appendix C and Appendix D include full dataset details, attribute distributions, network architecture, and experiment settings.

- `UTKFace` *(y: age, a: race)*: `UTKFace` is based on the UTKFace dataset (Zhang et al., 2017), which contains facial images annotated with age, gender, and ethnicity. We use the five ethnicity groups as the spurious attribute.
- `SkyFinder` *(y: temperature, a: camera ID)*: `SkyFinder` (Mihail et al., 2016) contains pixel-annotated time-lapse sky images captured by outdoor webcams. We use camera ID as the spurious attribute and predict the in-the-wild temperature associated with each image.
- `PovertyMap` *(y: poverty index, a: country)*: `PovertyMap` is based on PovertyMap-WILDS (Koh et al., 2021), which contains satellite images from rural and urban regions across multiple countries. We use country as the spurious attribute and poverty index as the regression target.
- `CodeNet` *(y: run time, a: language)*: `CodeNet` is based on the IBM Project CodeNet (Puri et al., 2021), which contains code submissions in multiple programming languages with metadata. We use CPU run time as the continuous target and programming language as the spurious attribute.

Baselines. We compare `L-MDS` and `F-MDS` with standard regression and debiasing baselines. These include ERM (Vapnik, 1998), classical inverse-frequency reweighting and square-root inverse reweighting, Class-Balanced Loss (Cui et al., 2019), and LDS (Yang et al., 2021), which smooths weights across nearby target values. To test whether classification-oriented spurious-correlation methods transfer to continuous prediction, we further evaluate DANN (Ganin et al., 2016); for `UTKFace`, `SkyFinder`, and `PovertyMap`, we also include GroupDRO (Sagawa et al., 2020) and RnC (Zha et al., 2023). This comparison covers standard ERM training, label-aware reweighting, regression imbalance methods, and representative spurious-correlation mitigation approaches.

Evaluation Process and Metrics. We evaluate each regression task using mean absolute error (MAE) and error geometric mean (GM) (Yang et al., 2021). To evaluate performance across different target-density regimes, we follow established long-tailed evaluation protocols (Yang et al., 2021; 2022) and divide target groups defined in Eqn. (1) into *many-shot*, *medium-shot*, *few-shot* regions, and *zero-shot* bins as those with no training samples. We report both overall and shot-region results.

Table 1: **Main results on `UTKFace`.** We report test MAE and its standard deviation across 5 random seeds.

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
ERM (Vapnik, 1998)	7.39 ±0.1	7.26 ±0.1	9.19 ±0.2	**4.34** ±0.2	18.61 ±2.3	6.24 ±0.1	19.18 ±0.7	7.19 ±0.1	13.16 ±1.9	9.83 ±0.2	73.56 ±7.3
Resample	7.64 ±0.1	7.50 ±0.1	9.40 ±0.3	4.72 ±0.1	19.71 ±1.2	6.20 ±0.1	17.12 ±0.9	7.00 ±0.3	13.03 ±1.4	10.43 ±0.4	76.58 ±7.1
SqrtReWeight	7.31 ±0.1	7.18 ±0.1	9.02 ±0.3	4.48 ±0.2	**16.30** ±2.3	6.10 ±0.1	**16.01** ±0.8	6.90 ±0.1	13.18 ±1.1	9.78 ±0.4	**62.10** ±8.2
ReWeight	8.42 ±0.1	8.25 ±0.1	9.62 ±0.2	6.21 ±0.1	17.82 ±0.9	7.06 ±0.1	18.00 ±1.4	7.65 ±0.1	14.78 ±1.4	10.88 ±0.3	86.59 ±4.6
CBLoss (Cui et al., 2019)	8.37 ±0.1	8.19 ±0.1	9.61 ±0.1	6.16 ±0.1	18.55 ±1.8	6.98 ±0.1	18.18 ±1.6	7.56 ±0.2	15.23 ±1.3	10.86 ±0.2	87.54 ±2.1
DANN (Ganin et al., 2016)	7.97 ±0.1	7.82 ±0.1	9.69 ±0.2	4.63 ±0.2	20.62 ±1.4	6.65 ±0.1	20.32 ±0.5	7.88 ±0.1	15.90 ±1.1	10.67 ±0.3	76.86 ±4.7
RnC (Zha et al., 2023)	7.38 ±0.1	7.25 ±0.1	9.22 ±0.2	4.35 ±0.1	19.31 ±1.1	6.15 ±0.0	17.64 ±1.3	7.12 ±0.1	12.55 ±0.9	9.91 ±0.2	63.69 ±6.2
LDS (Yang et al., 2021)	7.23 ±0.1	7.09 ±0.1	8.90 ±0.1	4.58 ±0.1	16.59 ±1.9	6.12 ±0.0	16.36 ±0.6	7.01 ±0.2	13.22 ±0.6	**9.46** ±0.3	71.49 ±10.6
GroupDRO (Sagawa et al., 2020)	7.43 ±0.1	7.29 ±0.1	9.02 ±0.1	4.94 ±0.1	16.69 ±1.7	6.13 ±0.1	17.78 ±1.5	6.86 ±0.2	12.71 ±0.9	9.90 ±0.1	71.79 ±11.7
`L-MDS`	7.30 ±0.1	7.17 ±0.1	8.94 ±0.2	4.66 ±0.2	19.40 ±2.1	**6.03** ±0.1	17.62 ±1.0	6.79 ±0.2	12.51 ±1.2	9.79 ±0.2	76.32 ±6.2
`F-MDS`	**7.22** ±0.1	**7.08** ±0.1	**8.71** ±0.2	4.65 ±0.2	17.49 ±1.9	6.08 ±0.1	17.91 ±0.2	**6.71** ±0.2	**12.43** ±1.4	9.54 ±0.4	68.81 ±8.1
`L-MDS + F-MDS`	7.42 ±0.1	7.29 ±0.1	9.04 ±0.2	4.55 ±0.2	17.55 ±1.8	6.15 ±0.1	17.73 ±1.8	7.02 ±0.1	12.88 ±0.6	9.96 ±0.3	77.43 ±9.3
Ours (best) vs. ERM	**+0.17**	**+0.18**	**+0.48**	**-0.21**	**+1.12**	**+0.21**	**+1.56**	**+0.48**	**+0.73**	**+0.29**	**+4.75**

Table 2: **Main results on `SkyFinder`.** We report test MAE and its standard deviation across 5 random seeds.

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
ERM (Vapnik, 1998)	3.68 ±0.0	3.41 ±0.0	5.95 ±0.1	**2.27** ±0.0	6.45 ±0.6	2.94 ±0.0	12.21 ±1.2	4.49 ±0.0	25.08 ±0.9	5.22 ±0.0	29.78 ±1.8
Resample	3.62 ±0.0	3.35 ±0.0	5.76 ±0.1	2.71 ±0.1	7.28 ±0.3	3.00 ±0.0	13.37 ±1.2	4.23 ±0.0	**19.23** ±0.6	4.97 ±0.1	36.23 ±4.0
SqrtReWeight	3.53 ±0.0	3.26 ±0.0	5.85 ±0.1	2.34 ±0.1	6.91 ±0.8	2.95 ±0.0	11.69 ±0.8	4.17 ±0.0	23.76 ±1.0	4.75 ±0.1	32.65 ±1.7
ReWeight	4.25 ±0.0	3.91 ±0.0	7.13 ±0.2	4.03 ±0.1	10.55 ±0.8	3.88 ±0.0	15.84 ±1.3	4.52 ±0.0	21.90 ±0.4	5.13 ±0.1	33.09 ±1.6
CBLoss (Cui et al., 2019)	4.23 ±0.1	3.90 ±0.1	7.24 ±0.2	4.07 ±0.2	10.39 ±0.8	3.86 ±0.1	15.04 ±2.1	4.50 ±0.1	20.16 ±0.6	5.12 ±0.1	30.88 ±1.0
DANN (Ganin et al., 2016)	4.04 ±0.1	3.76 ±0.1	6.75 ±0.2	2.57 ±0.0	7.98 ±0.7	3.32 ±0.1	12.60 ±1.3	4.83 ±0.1	24.84 ±0.4	5.56 ±0.1	31.02 ±1.6
RnC (Zha et al., 2023)	**3.49** ±0.0	**3.24** ±0.0	**5.69** ±0.1	2.42 ±0.0	7.21 ±0.5	**2.90** ±0.1	11.99 ±0.8	**4.14** ±0.0	19.65 ±1.2	**4.71** ±0.1	30.86 ±2.2
LDS (Yang et al., 2021)	3.85 ±0.1	3.56 ±0.0	6.44 ±0.4	2.39 ±0.0	7.95 ±0.6	3.12 ±0.0	13.51 ±1.0	4.69 ±0.1	21.71 ±0.9	5.26 ±0.1	33.98 ±2.8
GroupDRO (Sagawa et al., 2020)	3.62 ±0.0	3.35 ±0.0	6.00 ±0.1	2.34 ±0.0	6.64 ±0.5	**2.90** ±0.0	12.43 ±1.1	4.42 ±0.1	25.07 ±1.4	5.04 ±0.0	**29.66** ±1.5
`L-MDS`	3.54 ±0.0	3.27 ±0.0	5.81 ±0.2	2.38 ±0.0	6.86 ±0.6	2.95 ±0.0	**11.63** ±1.0	4.17 ±0.0	23.63 ±0.7	4.78 ±0.0	31.47 ±2.3
`F-MDS`	3.56 ±0.0	3.29 ±0.0	5.81 ±0.2	2.33 ±0.1	6.44 ±0.3	2.97 ±0.0	11.86 ±0.4	4.22 ±0.0	21.40 ±1.1	4.74 ±0.0	30.47 ±2.0
`L-MDS + F-MDS`	3.58 ±0.0	3.30 ±0.0	5.78 ±0.1	2.39 ±0.1	**6.28** ±0.6	2.97 ±0.0	12.18 ±0.6	4.23 ±0.1	22.30 ±1.3	4.82 ±0.1	32.99 ±2.8
Ours (best) vs. ERM	**+0.14**	**+0.14**	**+0.17**	**-0.06**	**+0.17**	**-0.01**	**+0.58**	**+0.32**	**+3.68**	**+0.48**	**-0.69**

Table 3: **Main results on `PovertyMap`.** We report test MAE and its standard deviation across 5 random seeds.

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
ERM (Vapnik, 1998)	0.504 ±0.0	0.502 ±0.0	0.679 ±0.0	**0.256** ±0.0	**0.504** ±0.1	0.335 ±0.0	1.356 ±0.1	0.494 ±0.0	2.452 ±0.1	0.744 ±0.0	1.996 ±0.1
Resample	0.506 ±0.0	0.503 ±0.0	0.710 ±0.0	0.385 ±0.0	0.781 ±0.2	0.391 ±0.0	1.383 ±0.1	**0.463** ±0.0	2.247 ±0.1	0.737 ±0.0	2.019 ±0.1
SqrtReWeight	0.512 ±0.0	0.509 ±0.0	0.670 ±0.0	0.375 ±0.0	0.679 ±0.2	0.376 ±0.0	1.441 ±0.1	0.478 ±0.0	2.233 ±0.1	0.753 ±0.0	2.037 ±0.1
ReWeight	0.522 ±0.0	0.520 ±0.0	0.750 ±0.0	0.485 ±0.1	0.805 ±0.1	0.431 ±0.0	1.426 ±0.1	0.464 ±0.0	2.088 ±0.2	0.748 ±0.0	2.012 ±0.1
CBLoss (Cui et al., 2019)	0.515 ±0.0	0.513 ±0.0	0.720 ±0.0	0.450 ±0.0	0.856 ±0.2	0.420 ±0.0	1.447 ±0.1	0.467 ±0.0	2.142 ±0.1	0.729 ±0.0	2.029 ±0.1
DANN (Ganin et al., 2016)	0.689 ±0.1	0.685 ±0.1	0.869 ±0.0	0.796 ±0.1	0.996 ±0.1	0.574 ±0.1	1.638 ±0.1	0.598 ±0.1	**1.926** ±0.1	1.003 ±0.1	2.191 ±0.1
RnC (Zha et al., 2023)	0.494 ±0.0	0.490 ±0.0	0.675 ±0.0	0.304 ±0.0	0.559 ±0.1	**0.290** ±0.0	**1.103** ±0.1	0.486 ±0.0	2.320 ±0.1	0.773 ±0.0	2.153 ±0.2
LDS (Yang et al., 2021)	0.501 ±0.0	0.499 ±0.0	0.712 ±0.0	0.331 ±0.0	0.717 ±0.1	0.336 ±0.0	1.458 ±0.1	0.501 ±0.0	2.276 ±0.1	**0.714** ±0.0	2.049 ±0.1
GroupDRO (Sagawa et al., 2020)	0.492 ±0.0	0.489 ±0.0	0.648 ±0.0	0.376 ±0.1	0.844 ±0.2	0.319 ±0.0	1.245 ±0.1	0.470 ±0.0	2.382 ±0.1	0.757 ±0.0	2.016 ±0.1
`L-MDS`	**0.486** ±0.0	**0.484** ±0.0	0.666 ±0.0	0.271 ±0.0	0.535 ±0.2	0.336 ±0.0	1.417 ±0.1	0.467 ±0.0	2.385 ±0.1	0.720 ±0.0	**1.987** ±0.1
`F-MDS`	0.488 ±0.0	0.485 ±0.0	0.670 ±0.0	0.278 ±0.0	0.554 ±0.1	0.327 ±0.0	1.307 ±0.1	0.477 ±0.0	2.492 ±0.1	0.719 ±0.0	2.057 ±0.0
`L-MDS + F-MDS`	0.492 ±0.0	0.490 ±0.0	**0.642** ±0.0	0.352 ±0.1	0.834 ±0.1	0.332 ±0.0	1.369 ±0.1	0.483 ±0.0	2.283 ±0.2	0.715 ±0.0	2.015 ±0.1
Ours (best) vs. ERM (%)	**+3.57%**	**+3.59%**	**+5.45%**	**-5.86%**	**-6.15%**	**+2.39%**	**+3.61%**	**+5.47%**	**+6.89%**	**+3.90%**	**+0.45%**

4.1 Main Results

We summarize the main results in this section for all DSR datasets and regression tasks. Additional results, training details, and hyperparameter settings can be found in Appendix A and D.

Age Regression Robust to Racial Attributes. Table 1 confirms that both `L-MDS` and `F-MDS` improve substantially over the ERM, with `F-MDS` achieving the best overall performance among all methods. The gains are especially clear in *few-shot* regions, where both methods outperform not only ERM, but also classification-oriented approaches such as DANN, GroupDRO, and RnC. Our methods provide a better balance between overall accuracy and robustness in sparse target regions.

Temperature Regression Robust to Camera Location. Table 2 reports results on `SkyFinder`, where camera ID is the spurious attribute. `L-MDS`, `F-MDS`, and their combination improve overall performance over ERM. The largest gains appear in the *few-shot* and *zero-shot* regions, where our methods reduce both average and worst-case MAE by substantial margins.

Table 4: **Main results on `CodeNet`.** We report test MAE and its standard deviation across 5 random seeds.

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
ERM (Vapnik, 1998)	268.7 ±2.7	269.0 ±2.7	350.3 ±11.3	123.3 ±2.6	354.5 ±48.4	221.9 ±4.0	667.9 ±40.3	371.9 ±4.6	716.6 ±11.7	565.8 ±6.2	937.5 ±6.0
ReWeight	253.7 ±2.8	253.5 ±2.5	306.3 ±9.1	174.4 ±3.3	445.5 ±77.8	203.9 ±4.6	660.4 ±84.4	302.0 ±5.7	709.4 ±69.8	503.2 ±7.4	844.5 ±3.9
SqrtReWeight	248.2 ±2.6	248.3 ±2.6	299.4 ±10.5	146.7 ±3.0	380.2 ±69.3	204.7 ±4.0	**428.9** ±59.8	312.2 ±4.9	641.1 ±55.0	498.1 ±7.3	959.0 ±12.0
CBLoss (Cui et al., 2019)	251.9 ±2.6	251.8 ±2.6	301.3 ±10.3	141.8 ±2.8	360.1 ±52.8	206.6 ±4.1	488.8 ±57.7	322.5 ±5.0	677.5 ±51.5	516.3 ±6.8	808.0 ±62.2
DANN (Ganin et al., 2016)	276.0 ±2.7	276.4 ±2.7	348.8 ±11.5	116.3 ±2.4	343.2 ±80.6	224.3 ±3.8	610.1 ±52.6	392.1 ±4.4	714.4 ±28.0	597.5 ±5.3	930.0 ±38.9
RnC (Zha et al., 2023)	292.0 ±2.8	292.4 ±2.8	356.5 ±11.5	124.5 ±2.4	381.7 ±94.0	255.7 ±4.4	663.5 ±33.0	401.4 ±4.5	737.6 ±55.6	615.3 ±5.9	931.0 ±31.8
LDS (Yang et al., 2021)	263.1 ±2.8	263.3 ±2.8	322.4 ±11.3	143.7 ±3.0	317.4 ±66.7	216.4 ±4.3	450.1 ±94.7	341.8 ±5.2	700.5 ±25.2	539.5 ±7.7	976.0 ±0.0
GroupDRO (Sagawa et al., 2020)	326.8 ±3.0	327.1 ±3.0	412.2 ±12.4	**110.6** ±2.1	556.8 ±13.2	316.7 ±4.2	791.5 ±18.2	467.8 ±4.1	879.7 ±28.0	661.5 ±5.1	976.0 ±0.0
Resample	256.3 ±2.6	256.1 ±2.6	298.8 ±10.4	158.7 ±3.1	355.9 ±47.9	215.0 ±4.3	532.0 ±65.6	317.1 ±5.1	636.4 ±61.3	497.3 ±6.7	799.5 ±27.9
`L-MDS`	**243.4** ±2.7	**243.1** ±2.7	299.0 ±9.3	127.1 ±2.8	**289.9** ±72.1	**192.6** ±4.2	519.0 ±33.4	318.7 ±5.2	635.1 ±52.8	523.6 ±7.2	823.0 ±0.0
`F-MDS`	250.5 ±2.5	250.4 ±2.5	**287.2** ±9.8	166.3 ±3.3	442.2 ±84.6	210.5 ±4.2	508.2 ±65.3	292.9 ±4.8	711.7 ±26.1	490.0 ±6.6	910.5 ±25.1
`L-MDS + F-MDS`	249.4 ±2.5	249.2 ±2.5	299.5 ±10.5	176.2 ±3.4	458.5 ±58.3	201.1 ±4.1	497.0 ±41.3	**288.5** ±4.7	**603.0** ±55.8	**483.4** ±7.2	**725.0** ±38.9
Ours (best) vs. ERM	**+25.3**	**+25.9**	**+63.1**	-3.7	**+64.6**	**+29.3**	**+170.9**	**+83.3**	**+113.6**	**+82.4**	**+212.5**

Poverty Index Regression Robust to Country. Table 3 reports `PovertyMap`, where country serves as the spurious attribute. All our variants improve over ERM overall, with `L-MDS` achieving the best overall performance among all compared methods. ERM shows clear signs of overfitting, whereas `L-MDS` and `F-MDS` generalize better across target regions with lower training prevalence.

Execution Time Regression Robust to Programming Language. For the LLM regression task trained on `CodeNet`, we verify in Table 4 that `L-MDS`, `F-MDS`, and their combination consistently improve overall performance, especially as data becomes sparser. `L-MDS` performs best overall, while `F-MDS` and `L-MDS + F-MDS` shine in few-shot and zero-shot regions.

Effectiveness of the Cross-Attribute MDS Term. To isolate the contribution of our cross-attribute smoothing, we compare our method against attribute-conditional LDS without the MDS attribute-axis term. Table 5 shows that our method significantly outperforms this baseline on all four datasets ($p < 0.05$), demonstrating that the cross-attribute MDS term provides improvements beyond target-axis smoothing alone.

Consistent Performance Across Datasets. Beyond individual datasets, the average performance ranking in Tables 6 and 19 shows that `L-MDS` and `F-MDS` consistently rank among the strongest methods across datasets and evaluation metrics. This indicates that their performance is not driven by a single dataset or target-density regime. We further evaluate other strong baselines, including FDS (Yang et al., 2021), JTT (Liu et al., 2021), and LfF (Nam et al., 2020), on all four datasets in Tables 11, 12, 13 and 14, where our method achieves consistent and superior performance against these additional strong baselines.

Table 5: **Cross-attribute MDS ablation.** Δ is the MAE improvement of our method over the baseline; $\Delta > 0$ favors our method.

Dataset	95% CI of Δ	p-value
`PovertyMap`	[0.003, 0.038]	**<0.05**
`UTKFace`	[0.024, 0.329]	**<0.05**
`SkyFinder`	[0.010, 0.103]	**<0.05**
`CodeNet`	[12.45, 21.73]	**<0.0001**

Table 6: **Average performance ranking across all datasets.** Full results for different metrics are in Table 19.

Rank	Method	Avg Rank
1	`L-MDS`	3.92
2	`F-MDS`	4.06
3	SqrtReWeight	4.88
4	`L-MDS + F-MDS`	5.20
5	RnC (Zha et al., 2023)	5.99
6	Resample	6.60
7	LDS (Yang et al., 2021)	6.63
8	ERM (Vapnik, 1998)	6.75
9	GroupDRO (Sagawa et al., 2020)	6.96
10	CBLoss (Cui et al., 2019)	8.31
11	ReWeight	8.94
12	DANN (Ganin et al., 2016)	9.80

4.2 Further Analysis

Ablation Studies for `L-MDS` & `F-MDS`. We study the robustness of `L-MDS` and `F-MDS` under several design choices. ❶ Since our main experiments use Gaussian smoothing, we vary the kernel size $k \in \{5, 9, 15\}$ and standard deviation $\sigma \in \{1, 2, 3\}$ to test sensitivity to smoothing strength. ❷ We compare Gaussian smoothing with alternative kernel types to evaluate whether performance depends on a specific kernel choice. ❸ We ablate the training loss function to examine whether the gains are tied to a particular regression objective. Across all settings, `L-MDS` and `F-MDS` remain robust and consistently outperform baselines. Detailed results are provided in Appendix B.2, B.3, B.4.

Interpolation & Extrapolation. Unlike classification, regression often requires predictions at target values that are unseen or missing during training. To test whether `L-MDS` and `F-MDS` can generalize to such zero-shot regions both within the training data coverage (i.e., *interpolation*) and outside of it (i.e., *extrapolation*), we curate a controlled subset of `UTKFace` by removing selected age intervals and truncating age extremes. While our main results evaluate naturally occurring missing regions,

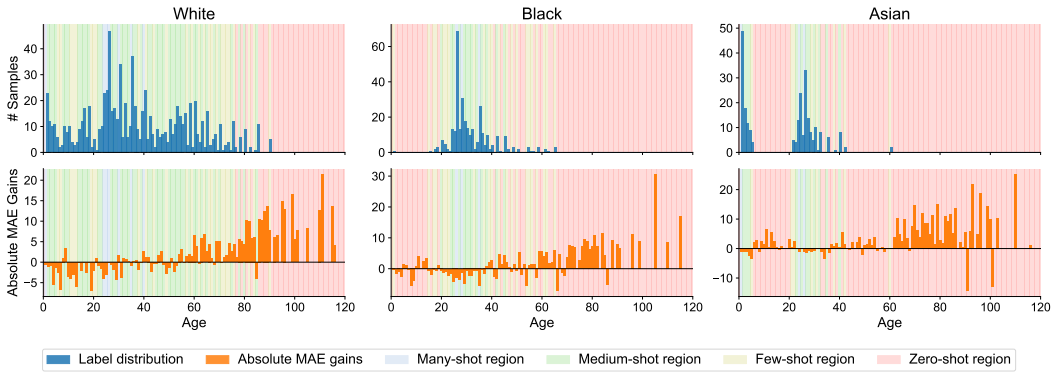

Figure 4: **Interpolation and extrapolation on zero-shot target regions.** We curate a `UTKFace` subset with missing age intervals and truncated age extremes. **Top:** per-attribute age distributions. **Bottom:** absolute MAE gains of `L-MDS` over ERM. Pink regions denote zero-shot target ranges, where `L-MDS` improves predictions despite no training samples. Detailed results for all attributes are provided in Appendix B.5.

this controlled setup allows us to isolate generalization to unseen target values. As detailed in Table 25, `L-MDS` substantially outperforms ERM in these zero-shot regions. Fig. 4 further visualizes the per-attribute age distributions and absolute MAE gains over ERM for three representative race groups, showing that `L-MDS` improves performance across both internal gaps and extrapolated age ranges. Detailed results for all attributes are provided in Appendix B.5.

Resilience to Reduced Training Data. Real-world datasets are often limited by sparse observations, annotation cost, and uneven target coverage. To test robustness under limited supervision, we subsample the `UTKFace` training set to 10%, 20%, 50%, and 100% of its original size, and compare `L-MDS` and `F-MDS` with the ERM baseline. As shown in Fig. 5, both `L-MDS` and `F-MDS` reduce performance degradation as the training set becomes smaller. Their gains over ERM are most clear in the low-data regimes, showing that attribute-aware smoothing is especially helpful when training coverage is sparse. Notably, `F-MDS` achieves the strongest improvement at 10% training data, suggesting that representation-based attribute smoothing can better recover transferable structure when direct supervision is limited. Detailed shot-wise results are provided in Appendix B.6.

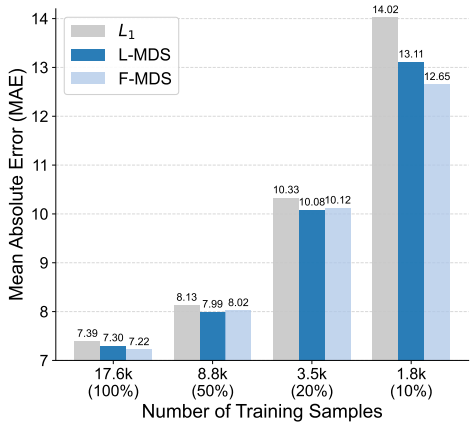

Figure 5: **Robustness to data scarcity.** We subsample `UTKFace` to simulate limited data settings. `L-MDS` and `F-MDS` obtain larger gains under lower data regimes.

5 Discussion

Limitations. While `L-MDS` and `F-MDS` boost overall performance, especially in *few-shot* and *zero-shot* regions, they may perform comparably to or slightly worse than baselines in some data-dense regions. This suggests a trade-off between preserving performance on well-represented target regions and improving robustness on sparse or unseen attribute-target combinations. In addition, our study focuses on settings where spurious attributes are available; future work may extend DSR to cases where such attributes are partially observed or unknown. Further impacts are stated in Appendix B.7.

Conclusion. We present DSR, a continuous prediction setting for studying spurious attribute-target correlations in deep regression. We introduce `L-MDS` and `F-MDS`, two MDS-based smoothing methods that exploit attribute similarity in label and feature spaces. Extensive experiments across vision, environmental sensing, and LLM-based regression show that our methods improve robustness in sparse target regions and achieve strong overall performance. Our work fills the gap in benchmarks and techniques for practical spurious correlation problems with continuous targets.

AI USE STATEMENT

In this work, we used generative AI tools for cleaning and filtering our datasets and basic data analysis tasks. We have not used generative AI tools for major methodology tasks, such as developing core conceptual frameworks, formulating mathematical claims, or implementing experiments. We have not used generative AI tools for generating synthetic data sets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, designing or providing feedback on research, implementing methods, and interpreting results. All data analysis scripts and the results were verified by at least two authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work strictly follows the ICLR Code of Ethics. We did not collect any new data for our experiments, and none of our experiments directly involve human participants. All datasets involved in the paper are publicly available to download. Only one dataset, `UTKFace`, involves images of human faces, but is formally licensed for academic research purposes. Our paper's contributions are primarily methodological, and we believe that it can positively impact the field of machine learning. Our work can greatly improve continuous prediction systems, such as medical risk prediction and poverty estimation. However, some risks may entail use cases where a false sense of robustness exacerbates discrimination if predictions are used for purposes like surveillance or screening without proper oversight. We have not deployed our methods in such contexts. We prioritized reporting all results honestly, acknowledging limitations, and following best practices for research integrity. No conflicts of interest or ethical concerns exist beyond standard research conduct.

REPRODUCIBILITY STATEMENT

We have followed every step toward ensuring the reproducibility of both our methods and results. The paper details the datasets used, network architecture, baseline comparison algorithms, and evaluation metrics in Section 4. Appendix D contains the full experiment settings, such as hyperparameter choices, type of optimizer, and other training details. All source code and data will be made publicly available upon acceptance, together with instructions for reproduction. All resources provided ensure that independent researchers can both verify and reproduce our methods and findings.

REFERENCES

Yash Akhauri, Xingyou Song, Arissa Wongpanich, Bryan Lewandowski, and Mohamed S Abdelfattah. Regression language models for code. *arXiv preprint arXiv:2509.26476*, 2025.

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *European conference on computer vision*, pp. 472–489. Springer, 2018.

Ingwer Borg and Patrick J.F. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

Paula Branco, Luís Torgo, and Rita P. Ribeiro. A survey of predictive modelling under imbalanced domains. *ACM Computing Surveys*, 49(2):1–50, 2016.

Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, pp. 77–91. PMLR, 2018.

Elliot Creager, Jörn-Henrik Jacobsen, and Richard Zemel. Environment inference for invariant learning. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 2189–2200. PMLR, 2021.

Imre Csiszár and Paul C. Shields. Information theory and statistics: A tutorial. *Foundations and Trends in Communications and Information Theory*, 1(4):417–528, 2004.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 9260–9269. IEEE, 2019.

John C Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.

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.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 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.

Gregory Holste, Yiliang Zhou, Song Wang, Ajay Jaiswal, Mingquan Lin, Sherry Zhuge, Yuzhe Yang, Dongkyun Kim, Trong-Hieu Nguyen-Mau, Minh-Triet Tran, Jaehyup Jeong, Wongi Park, Jongbin Ryu, Feng Hong, Arsh Verma, Yosuke Yamagishi, Changhyun Kim, Hyeryeong Seo, Myungjoo Kang, Leo Anthony Celi, Zhiyong Lu, Ronald M. Summers, George Shih, Zhangyang Wang, and Yifan Peng. Towards long-tailed, multi-label disease classification from chest x-ray: Overview of the cxr-lt challenge. *Medical Image Analysis*, 97:103224, 2024.

Zhiwen Hu, Zixuan Bai, Yuzhe Yang, Zijie Zheng, Kaigui Bian, and Lingyang Song. Uav aided aerial-ground iot for air quality sensing in smart city: Architecture, technologies, and implementation. *IEEE Network*, 33(2):14–22, 2019.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Learning Representations*, 2023.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 5637–5664. PMLR, 2021.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Evan Z. Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just train twice: Improving group robustness without training group information. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 6781–6792. PMLR, 2021.

Radu Paul Mihail, Scott Workman, Zach Bessinger, and Nathan Jacobs. Sky segmentation in the wild: An empirical study. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1–6, 2016.

Junhyun Nam, Hyuntak Cha, Sungsoo Ahn, Jaeho Lee, and Jinwoo Shin. Learning from failure: Training debiased classifier from biased classifier. In *Advances in Neural Information Processing Systems*, volume 33, pp. 20673–20684, 2020.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends in Machine Learning*, 11(5–6):355–607, 2019.

Ruchir Puri, David S. Kung, Geert Janssen, Wei Zhang, Giacomo Domeniconi, Vladimir Zolotov, Julian Dolby, Jie Chen, Mihir Choudhury, Lindsey Decker, Veronika Thost, Luca Buratti, Saurabh Pujar, Shyam Ramji, Ulrich Finkler, Susan Malaika, and Frederick Reiss. Codenet: A large-scale ai for code dataset for learning a diversity of coding tasks. *arXiv preprint arXiv:2105.12655*, 2021.

Jiawei Ren, Mingyuan Zhang, Cunjun Yu, and Ziwei Liu. Balanced MSE for imbalanced visual regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7926–7935, 2022.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121, 2000.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9573–9585, 2020.

Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Grundlehren der mathematischen Wissenschaften*. Springer, Berlin, Heidelberg, 2009.

Yuzhe Yang, Kaiwen Zha, Ying-Cong Chen, Hao Wang, and Dina Katabi. Delving into deep imbalanced regression. In *International Conference on Machine Learning (ICML)*, pp. 11842–11851. PMLR, 2021.

Yuzhe Yang, Hao Wang, and Dina Katabi. On multi-domain long-tailed recognition, imbalanced domain generalization and beyond. In *European conference on computer vision*, pp. 57–75. Springer, 2022.

Yuzhe Yang, Xin Liu, Jiang Wu, Silviu Borac, Dina Katabi, Ming-Zher Poh, and Daniel McDuff. Simper: Simple self-supervised learning of periodic targets. In *International Conference on Learning Representations (ICLR)*, 2023a.

Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. Change is hard: A closer look at subpopulation shift. *International Conference on Machine Learning (ICML)*, 2023b.

Yuzhe Yang, Haoran Zhang, Judy W. Gichoya, Dina Katabi, and Marzyeh Ghassemi. The limits of fair medical imaging ai in real-world generalization. *Nature Medicine*, 30:2838–2848, 2024.

Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. Rank-N-contrast: Learning continuous representations for regression. *Advances in Neural Information Processing Systems*, 36:17882–17903, 2023.

Michael Zhang, Nimit Sharad Sohoni, Hongyang R. Zhang, Chelsea Finn, and Christopher Ré. Correct-n-contrast: A contrastive approach for improving robustness to spurious correlations. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 26484–26516. PMLR, 2022.

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 (CVPR)*, pp. 5810–5818, 2017.

A Additional Results

We report the complete evaluation results on all four datasets using error geometric mean (GM) (Yang et al., 2021), which serves as a supplement to the provided results in the main paper. As a whole, `L-MDS` and `F-MDS` perform better than the baseline and other methods when evaluated with GM rather than MAE, indicating more uniform accuracy across attributes.

A.1 GM results on UTKFace

We provide GM evaluation results for `UTKFace` in Table 7. As the table illustrates, our method yields consistent improvements over ERM across most GM metrics on `UTKFace` with the most pronounced gains in the *few-shot* and *zero-shot* regions. Notably, `F-MDS` achieves the best overall GM and ranks among the top methods in both attribute-level and shot-based evaluations, confirming the effectiveness of our method across diverse evaluation criteria.

Table 7: Additional `UTKFace` results. We report test GM and its standard deviation across 5 random seeds.

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
ERM (Vapnik, 1998)	4.01 ±0.1	4.02 ±0.1	5.32 ±0.1	**2.18** ±0.2	9.23 ±1.3	3.57 ±0.1	11.64 ±0.7	4.46 ±0.1	10.61 ±0.3	5.62 ±0.1	73.56 ±7.2
Resample	4.15 ±0.1	4.15 ±0.1	5.37 ±0.3	2.39 ±0.1	10.22 ±1.5	3.52 ±0.1	11.62 ±0.9	4.25 ±0.3	10.35 ±0.9	6.08 ±0.3	76.58 ±7.1
SqrtReWeight	4.01 ±0.0	4.01 ±0.0	5.18 ±0.2	2.27 ±0.2	8.91 ±0.4	3.47 ±0.1	11.12 ±1.0	4.35 ±0.1	11.23 ±1.5	5.72 ±0.2	**62.10** ±8.2
ReWeight	5.08 ±0.3	5.02 ±0.3	5.91 ±0.3	3.34 ±0.3	10.35 ±1.4	4.36 ±0.3	14.19 ±1.4	5.23 ±0.3	12.37 ±0.8	6.95 ±0.4	87.63 ±3.8
CBLoss (Cui et al., 2019)	4.97 ±0.2	4.91 ±0.2	5.78 ±0.3	3.29 ±0.2	10.34 ±1.1	4.24 ±0.2	13.52 ±1.5	5.01 ±0.3	12.40 ±1.0	6.85 ±0.4	87.48 ±2.4
DANN (Ganin et al., 2016)	4.34 ±0.1	4.33 ±0.1	5.60 ±0.2	2.25 ±0.2	11.94 ±1.0	3.83 ±0.1	13.13 ±1.1	5.00 ±0.1	11.59 ±0.8	6.17 ±0.2	76.86 ±4.7
RnC (Zha et al., 2023)	4.07 ±0.1	4.08 ±0.1	5.34 ±0.1	2.30 ±0.1	8.48 ±0.9	3.55 ±0.1	11.72 ±1.6	4.46 ±0.1	10.32 ±1.0	5.72 ±0.2	62.91 ±6.6
LDS (Yang et al., 2021)	3.95 ±0.1	3.94 ±0.1	5.09 ±0.1	2.35 ±0.1	**8.40** ±0.6	3.45 ±0.0	**11.04** ±0.6	4.41 ±0.2	10.56 ±0.8	**5.40** ±0.2	71.10 ±11.3
GroupDRO (Sagawa et al., 2020)	4.08 ±0.1	4.06 ±0.1	5.13 ±0.1	2.64 ±0.1	9.63 ±0.6	3.46 ±0.0	12.40 ±1.6	4.14 ±0.2	9.95 ±0.8	5.70 ±0.1	71.79 ±11.7
`L-MDS`	3.95 ±0.0	3.94 ±0.0	5.12 ±0.1	2.44 ±0.1	9.94 ±1.6	**3.35** ±0.1	12.25 ±1.9	4.10 ±0.2	10.10 ±0.8	5.58 ±0.2	76.32 ±6.2
`F-MDS`	**3.91** ±0.1	**3.90** ±0.1	**4.90** ±0.2	2.40 ±0.1	8.94 ±0.7	3.40 ±0.0	12.56 ±0.7	**4.03** ±0.2	**9.58** ±0.5	5.43 ±0.3	68.81 ±8.0
`L-MDS + F-MDS`	4.06 ±0.1	4.05 ±0.0	5.09 ±0.1	2.31 ±0.1	8.66 ±0.6	3.53 ±0.1	11.63 ±1.0	4.37 ±0.1	10.40 ±0.5	5.75 ±0.2	77.43 ±9.3
Ours (best) vs. ERM	**+2.5%**	**+3.0%**	**+7.9%**	**-6.0%**	**+6.2%**	**+6.2%**	**+0.1%**	**+9.6%**	**+9.7%**	**+3.4%**	**+6.5%**

A.2 GM Results on SkyFinder

Complete GM results on `SkyFinder` are shown in Table 8. Similar to `UTKFace`, our method demonstrates stronger performance in the *few-shot* and *zero-shot* regions, indicating improved robustness for underrepresented target groups where standard ERM typically struggles. At the same time, the method remains competitive on the medium-shot and many-shot regions, showing that the gains do not come at the cost of performance on well-represented samples. We also observe that RnC (Zha et al., 2023) serves as a strong and competitive baseline on this benchmark dataset, and our method achieves comparable overall performance.

Table 8: Additional `SkyFinder` results. We report test GM and its standard deviation across 5 random seeds.

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
ERM (Vapnik, 1998)	2.26 ±0.0	2.19 ±0.0	3.71 ±0.1	**1.38** ±0.0	6.00 ±0.8	1.84 ±0.0	10.92 ±1.2	2.86 ±0.0	22.46 ±0.6	3.47 ±0.1	29.78 ±1.8
Resample	2.23 ±0.0	2.14 ±0.0	3.82 ±0.2	1.69 ±0.1	6.78 ±0.5	1.85 ±0.0	10.62 ±0.9	2.68 ±0.0	**17.47** ±0.7	3.38 ±0.1	36.23 ±4.0
SqrtReWeight	**2.17** ±0.0	**2.09** ±0.0	3.79 ±0.2	1.44 ±0.1	6.59 ±1.0	1.83 ±0.0	10.92 ±0.9	2.63 ±0.1	21.04 ±0.8	3.14 ±0.1	32.65 ±1.7
ReWeight	2.68 ±0.0	2.54 ±0.0	4.66 ±0.2	2.53 ±0.1	10.04 ±1.0	2.44 ±0.0	14.19 ±2.1	2.83 ±0.1	19.66 ±0.8	3.42 ±0.1	33.09 ±1.6
CBLoss (Cui et al., 2019)	2.67 ±0.1	2.53 ±0.1	4.76 ±0.1	2.65 ±0.2	9.77 ±0.4	2.44 ±0.1	13.81 ±1.8	2.81 ±0.1	17.77 ±1.2	3.33 ±0.1	30.88 ±1.0
DANN (Ganin et al., 2016)	2.53 ±0.0	2.44 ±0.0	4.55 ±0.1	1.52 ±0.1	7.40 ±0.9	2.12 ±0.1	12.09 ±1.3	3.14 ±0.0	22.99 ±0.4	3.68 ±0.1	31.02 ±1.6
RnC (Zha et al., 2023)	**2.17** ±0.0	2.10 ±0.0	**3.68** ±0.2	1.56 ±0.1	7.08 ±0.5	1.83 ±0.1	10.50 ±0.9	**2.62** ±0.0	18.19 ±1.3	**3.03** ±0.1	30.86 ±2.2
LDS (Yang et al., 2021)	2.38 ±0.0	2.29 ±0.0	4.00 ±0.2	1.41 ±0.0	7.63 ±0.7	1.96 ±0.0	11.36 ±0.5	3.05 ±0.1	20.84 ±1.0	3.48 ±0.1	33.98 ±2.8
GroupDRO (Sagawa et al., 2020)	2.22 ±0.0	2.14 ±0.0	3.76 ±0.1	1.42 ±0.1	6.48 ±0.5	**1.79** ±0.0	11.00 ±1.6	2.85 ±0.0	23.21 ±1.0	3.33 ±0.0	**29.66** ±1.5
`L-MDS`	2.19 ±0.0	2.10 ±0.0	3.72 ±0.2	1.44 ±0.1	6.46 ±0.5	1.85 ±0.0	**10.00** ±0.9	**2.62** ±0.0	20.97 ±0.7	3.22 ±0.1	31.47 ±2.3
`F-MDS`	2.18 ±0.0	**2.09** ±0.0	3.75 ±0.1	1.40 ±0.0	6.11 ±0.5	1.84 ±0.0	10.36 ±1.2	2.70 ±0.1	18.53 ±0.9	3.09 ±0.0	30.47 ±2.0
`L-MDS + F-MDS`	2.20 ±0.0	2.10 ±0.0	**3.68** ±0.1	1.47 ±0.1	**5.90** ±0.4	1.85 ±0.0	10.83 ±0.7	2.64 ±0.1	20.13 ±1.4	3.20 ±0.1	32.99 ±2.8
Ours (best) vs. ERM	**+3.5%**	**+4.6%**	**+0.8%**	**-1.4%**	**+1.7%**	**+0.0%**	**+8.4%**	**+8.4%**	**+17.5%**	**+11.0%**	**-2.3%**

A.3 GM Results on PovertyMap

Table 9 contains all GM metrics calculated from evaluating the `PovertyMap` dataset. When comparing performance with GM, `L-MDS` and `F-MDS` actually improve on the baseline across all shot regions, as opposed to a slight degradation in the *many-shot* region when evaluating with MAE. The strong GM metrics for `L-MDS` and `F-MDS` suggest that both methods balance out prediction errors across the entire distribution, while the baseline instead focuses on reducing error in *many-shot* regions.

Table 9: Additional `PovertyMap` results. We report test GM and its standard deviation across 5 random seeds.

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
ERM (Vapnik, 1998)	0.33 ±0.0	0.34 ±0.0	0.50 ±0.0	0.20 ±0.0	**0.50** ±0.1	0.21 ±0.0	1.36 ±0.1	0.34 ±0.0	2.45 ±0.1	0.59 ±0.0	2.00 ±0.1
Resample	0.33 ±0.0	0.34 ±0.0	0.54 ±0.0	0.26 ±0.1	0.78 ±0.2	0.25 ±0.0	1.38 ±0.1	**0.30** ±0.0	2.25 ±0.1	0.56 ±0.0	2.02 ±0.1
SqrtReWeight	0.33 ±0.0	0.34 ±0.0	0.49 ±0.0	0.28 ±0.1	0.68 ±0.2	0.23 ±0.0	1.44 ±0.1	0.32 ±0.0	2.23 ±0.1	0.57 ±0.0	2.04 ±0.1
ReWeight	0.34 ±0.0	0.35 ±0.0	0.58 ±0.0	0.33 ±0.1	0.80 ±0.1	0.29 ±0.0	1.43 ±0.1	**0.30** ±0.0	2.09 ±0.2	**0.53** ±0.0	2.01 ±0.1
CBLoss (Cui et al., 2019)	0.33 ±0.0	0.34 ±0.0	0.55 ±0.0	0.31 ±0.0	0.86 ±0.2	0.28 ±0.0	1.45 ±0.1	**0.30** ±0.0	2.14 ±0.1	**0.53** ±0.0	2.03 ±0.1
DANN (Ganin et al., 2016)	0.48 ±0.1	0.48 ±0.1	0.63 ±0.0	0.78 ±0.1	1.00 ±0.1	0.41 ±0.1	1.64 ±0.1	0.41 ±0.0	**1.93** ±0.1	0.76 ±0.1	2.19 ±0.1
RnC (Zha et al., 2023)	**0.32** ±0.0	**0.32** ±0.0	0.46 ±0.0	0.26 ±0.0	0.56 ±0.1	**0.19** ±0.0	**1.10** ±0.1	0.33 ±0.0	2.32 ±0.1	0.61 ±0.0	2.15 ±0.2
LDS (Yang et al., 2021)	0.33 ±0.0	0.33 ±0.0	0.51 ±0.0	0.22 ±0.1	0.72 ±0.1	0.20 ±0.0	1.46 ±0.1	0.34 ±0.0	2.28 ±0.1	0.56 ±0.0	2.05 ±0.1
GroupDRO (Sagawa et al., 2020)	**0.32** ±0.0	0.33 ±0.0	**0.45** ±0.0	0.26 ±0.1	0.84 ±0.2	0.20 ±0.0	1.24 ±0.1	0.33 ±0.0	2.38 ±0.1	0.58 ±0.0	2.02 ±0.1
`L-MDS`	**0.32** ±0.0	**0.32** ±0.0	0.49 ±0.0	0.19 ±0.1	0.54 ±0.2	0.20 ±0.0	1.42 ±0.1	0.32 ±0.0	2.39 ±0.1	0.57 ±0.0	**1.99** ±0.1
`F-MDS`	**0.32** ±0.0	**0.32** ±0.0	0.48 ±0.0	**0.17** ±0.0	0.55 ±0.1	0.20 ±0.0	1.31 ±0.1	0.33 ±0.0	2.49 ±0.1	0.55 ±0.0	2.06 ±0.0
`L-MDS + F-MDS`	**0.32** ±0.0	**0.32** ±0.0	0.46 ±0.0	0.19 ±0.1	0.83 ±0.1	0.20 ±0.0	1.37 ±0.1	0.33 ±0.0	2.28 ±0.2	0.55 ±0.0	2.02 ±0.1
Ours (best) vs. ERM	**+3.0%**	**+5.9%**	**+8.0%**	**+15.0%**	**-8.0%**	**+4.8%**	**+3.7%**	**+5.9%**	**+6.9%**	**+6.8%**	**+0.5%**

A.4 GM Results on CodeNet

We report all GM results for the `CodeNet` dataset in Table 10. We observe that GM metrics for `L-MDS`, `F-MDS`, and `L-MDS` and `F-MDS` combined further amplify the MAE results discussed in the main paper. As demonstrated with MAE, all three methods improve the baseline performance and make substantial gains in *medium-shot* and *few-shot* regions. The test GM results additionally demonstrate the dominance of `L-MDS`, where it beats all other methods and is directly comparable to the baseline, even in *many-shot* regions.

Table 10: Additional `CodeNet` results. We report test GM and its standard deviation across 5 random seeds.

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
ERM (Vapnik, 1998)	155.9 ±2.6	158.7 ±2.6	217.6 ±12.4	65.7 ±1.9	319.3 ±44.8	142.6 ±4.1	656.9 ±40.1	300.7 ±6.0	715.6 ±11.4	531.4 ±7.9	937.5 ±6.0
ReWeight	146.1 ±2.4	148.7 ±2.4	207.8 ±10.7	99.5 ±2.8	376.6 ±71.4	113.1 ±3.5	561.7 ±143.2	**188.6** ±5.6	662.5 ±72.2	448.4 ±9.3	844.5 ±3.9
SqrtReWeight	142.6 ±2.4	144.4 ±2.4	185.0 ±10.3	77.3 ±2.4	298.2 ±60.8	123.9 ±3.7	379.0 ±42.8	219.4 ±5.7	618.5 ±61.8	437.5 ±9.9	958.8 ±12.0
CBLoss (Cui et al., 2019)	144.0 ±2.4	145.7 ±2.4	185.6 ±10.4	78.0 ±2.3	299.2 ±58.8	120.3 ±3.8	436.5 ±70.1	227.0 ±6.0	655.9 ±54.9	459.6 ±10.2	803.2 ±62.1
DANN (Ganin et al., 2016)	165.8 ±2.6	168.3 ±2.6	213.6 ±12.4	65.6 ±1.8	255.7 ±48.5	155.5 ±3.9	579.5 ±69.7	336.0 ±5.5	710.2 ±29.9	574.3 ±6.7	928.4 ±38.9
RnC (Zha et al., 2023)	181.5 ±2.8	182.7 ±2.8	217.9 ±12.8	73.1 ±2.0	282.0 ±29.7	182.3 ±4.4	654.7 ±34.4	345.5 ±5.6	716.2 ±41.7	588.2 ±7.3	929.9 ±31.8
LDS (Yang et al., 2021)	146.2 ±2.5	149.1 ±2.5	187.8 ±10.8	73.3 ±2.3	261.2 ±50.4	128.2 ±4.0	**362.5** ±46.2	241.5 ±6.4	695.1 ±26.6	470.8 ±11.3	976.0 ±0.0
GroupDRO (Sagawa et al., 2020)	206.2 ±3.2	208.1 ±3.2	258.0 ±15.4	65.0 ±1.8	555.1 ±13.5	261.9 ±4.4	789.0 ±18.0	432.2 ±4.6	877.0 ±27.5	637.7 ±8.4	976.0 ±0.0
Resample	150.8 ±2.5	152.3 ±2.5	182.3 ±10.3	88.8 ±2.6	323.3 ±42.5	127.3 ±3.9	413.8 ±58.9	218.1 ±5.9	607.0 ±59.6	448.3 ±8.8	798.5 ±27.9
`L-MDS`	**124.5** ±2.3	**127.1** ±2.3	**171.6** ±10.9	**63.4** ±2.0	**224.1** ±47.3	**100.6** ±3.5	510.0 ±34.2	205.8 ±6.3	610.7 ±48.4	460.4 ±10.7	823.0 ±0.0
`F-MDS`	147.4 ±2.4	148.7 ±2.4	178.1 ±9.8	88.6 ±2.7	273.6 ±128.8	127.8 ±3.8	462.3 ±72.1	200.3 ±5.4	706.7 ±25.2	444.6 ±8.3	909.8 ±25.1
`L-MDS + F-MDS`	147.4 ±2.4	149.1 ±2.4	185.6 ±10.1	97.4 ±2.8	416.8 ±37.6	115.9 ±3.6	484.4 ±42.1	200.7 ±5.3	**566.3** ±70.4	**423.3** ±9.7	**722.9** ±38.9
Ours (best) vs. ERM	**+31.4**	**+31.6**	**+46.0**	**+2.3**	**+95.3**	**+42.0**	**+194.6**	**+100.4**	**+149.3**	**+108.2**	**+214.5**

A.5 Additional Baseline Comparisons

We further compare our method against three additional strong baselines: FDS (Yang et al., 2021), JTT (Liu et al., 2021), and LfF (Nam et al., 2020). These methods provide complementary comparisons to feature-space regression rebalancing and label-free spurious-correlation mitigation. We follow the original implementations as closely as possible and evaluate all methods using the same metrics as in the main experiments. Table 11, 12, 13 report the results on `UTKFace`, `SkyFinder`, `PovertyMap` respectively. Our method achieves the best overall MAE compared with FDS, JTT,

Table 11: Additional baseline results on `UTKFace`. We report test MAE and its standard deviation across 5 random seeds.

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
FDS (Yang et al., 2021)	7.61 ±0.1	7.48 ±0.1	9.23 ±0.1	4.50 ±0.2	18.57 ±1.6	6.31 ±0.1	17.68 ±0.8	7.51 ±0.4	15.00 ±1.7	10.19 ±0.1	76.20 ±10.0
JTT (Liu et al., 2021)	7.85 ±0.2	7.71 ±0.2	9.68 ±0.2	4.68 ±0.2	17.63 ±1.3	6.44 ±0.1	18.96 ±0.7	7.55 ±0.2	14.26 ±1.7	10.62 ±0.3	75.84 ±12.3
LfF (Nam et al., 2020)	7.40 ±0.1	7.26 ±0.1	9.22 ±0.3	**4.41** ±0.1	17.81 ±1.5	6.16 ±0.1	**16.88** ±1.3	7.07 ±0.2	13.42 ±1.7	9.93 ±0.2	**61.91** ±12.3
Ours	**7.22** ±0.1	**7.08** ±0.1	**8.71** ±0.2	4.65 ±0.1	**17.49** ±1.9	**6.08** ±0.1	17.91 ±0.2	**6.71** ±0.1	**12.43** ±1.4	**9.54** ±0.4	68.81 ±8.1

Table 12: Additional baseline results on `SkyFinder`. We report test MAE and its standard deviation across 5 random seeds.

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
FDS (Yang et al., 2021)	3.77 ±0.1	3.51 ±0.0	6.11 ±0.1	2.40 ±0.1	7.52 ±0.8	3.07 ±0.1	11.96 ±1.0	4.55 ±0.1	**22.22** ±0.5	5.21 ±0.1	31.61 ±1.2
JTT (Liu et al., 2021)	3.93 ±0.1	3.63 ±0.1	6.49 ±0.2	2.69 ±0.1	7.26 ±0.6	3.30 ±0.0	12.41 ±0.9	4.67 ±0.1	26.38 ±2.0	5.09 ±0.1	31.63 ±2.3
LfF (Nam et al., 2020)	3.66 ±0.0	3.40 ±0.0	6.17 ±0.1	**2.29** ±0.0	**5.51** ±0.4	2.96 ±0.0	12.42 ±1.8	4.45 ±0.1	24.53 ±0.8	5.09 ±0.1	33.10 ±1.1
Ours	**3.54** ±0.0	**3.27** ±0.0	**5.81** ±0.2	2.38 ±0.0	6.86 ±0.6	**2.95** ±0.0	**11.63** ±1.0	**4.17** ±0.0	23.63 ±0.7	**4.78** ±0.0	**31.47** ±2.3

Table 13: Additional baseline results on `PovertyMap`. We report test MAE and its standard deviation across 5 random seeds.

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
FDS (Yang et al., 2021)	0.500 ±0.0	0.498 ±0.0	0.684 ±0.0	**0.258** ±0.0	**0.449** ±0.1	**0.321** ±0.0	**1.346** ±0.1	0.498 ±0.0	2.526 ±0.1	0.737 ±0.0	2.032 ±0.1
JTT (Liu et al., 2021)	0.516 ±0.0	0.514 ±0.0	0.695 ±0.0	0.285 ±0.1	0.566 ±0.2	0.360 ±0.0	1.460 ±0.1	0.534 ±0.0	2.570 ±0.2	**0.686** ±0.0	2.069 ±0.1
LfF (Nam et al., 2020)	0.522 ±0.0	0.518 ±0.0	0.691 ±0.0	0.353 ±0.2	0.597 ±0.2	0.348 ±0.1	1.496 ±0.2	0.507 ±0.0	2.830 ±0.8	0.774 ±0.0	2.069 ±0.2
Ours	**0.486** ±0.0	**0.484** ±0.0	**0.666** ±0.0	0.271 ±0.0	0.535 ±0.2	0.336 ±0.0	1.417 ±0.1	**0.467** ±0.0	**2.385** ±0.1	0.720 ±0.0	**1.987** ±0.1

Table 14: Additional baseline results on `CodeNet`. We report test MAE and its standard deviation across 5 random seeds.

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
FDS (Yang et al., 2021)	275.4 ±3.0	275.5 ±2.9	333.0 ±8.6	124.5 ±5.6	338.5 ±58.7	232.9 ±3.6	551.1 ±18.7	378.6 ±7.5	715.6 ±19.8	578.6 ±15.1	940.5 ±26.6
JTT (Liu et al., 2021)	256.7 ±6.8	256.5 ±6.7	**287.8** ±5.3	192.5 ±18.8	484.9 ±54.8	207.6 ±4.8	528.6 ±33.4	**294.1** ±11.3	636.3 ±73.5	**470.6** ±22.1	**762.8** ±26.1
LfF (Nam et al., 2020)	274.6 ±5.4	274.7 ±5.5	331.0 ±21.1	**123.7** ±4.9	361.0 ±82.7	232.8 ±3.2	566.7 ±47.5	378.2 ±9.0	717.3 ±25.0	575.2 ±15.7	957.8 ±31.9
Ours	**243.4** ±2.7	**243.1** ±2.7	299.0 ±9.3	127.1 ±2.8	**289.9** ±72.1	**192.6** ±4.2	**519.0** ±33.4	318.7 ±5.2	**635.1** ±52.8	523.6 ±7.2	823.0 ±0.0

and LfF, while also showing strong performance across attribute- and shot-based metrics. These additional comparisons further support the consistent performance of our method across diverse DSR settings.

B Further Analysis & Ablation Studies

B.1 Average Metric Rank

In this section, we provide the average ranking of methods across metrics for each dataset separately. Average ranking for `UTKFace`, `SkyFinder`, `PovertyMap`, `CodeNet` are presented in Table 15, Table 16, Table 17 and Table 18 respectively. For each metric, methods are ranked based on their performance, where rank 1 corresponds to the best-performing method. We then compute the average rank of each method across all reported metrics within a dataset. The ranking includes overall metrics as well as group-wise metrics across many-shot, medium-shot, few-shot, and zero-shot regions. Our method achieves the best average ranking on `UTKFace`, `PovertyMap`, and `CodeNet`, while remaining highly competitive on `SkyFinder`. These results suggest that our method provides consistently strong performance across diverse evaluation metrics and datasets, demonstrating strong generalization ability across diverse task domains. We also expand Table 6 to report the average rank for each metric.

Table 15: Average ranking of methods across metrics on `UTKFace`. Lower average rank is better. Rankings are computed over the 11 MAE metrics reported in the main table.

Rank	Method	Average Metric Rank
1	`F-MDS`	2.82
2	SqrtReWeight	3.32
3	LDS	3.55
4	`L-MDS`	4.27
5	RnC	5.50
6	GroupDRO	5.73
7	`L-MDS` + `F-MDS`	6.55
8	ERM	6.82
9	Resample	7.82
10	DANN	10.45
10	CBLoss	10.45
12	ReWeight	10.73

Table 16: Average ranking of methods across metrics on `SkyFinder`. Lower average rank is better. Rankings are computed over the 11 MAE metrics reported in the main table.

Rank	Method	Average Metric Rank
1	RnC	2.86
2	`F-MDS`	3.55
3	`L-MDS`	4.32
4	SqrtReWeight	4.41
5	`L-MDS` + `F-MDS`	5.32
6	GroupDRO	5.73
7	ERM	6.18
8	Resample	6.77
9	LDS	9.05
10	CBLoss	9.45
11	DANN	9.73
12	ReWeight	10.64

Table 17: Average ranking of methods across metrics on `PovertyMap`. Lower average rank is better. Rankings are computed over the 11 MAE metrics reported in the main table.

Rank	Method	Average Metric Rank
1	`L-MDS`	3.73
2	`F-MDS`	4.68
3	`L-MDS` + `F-MDS`	4.91
4	GroupDRO	5.41
5	ERM	5.64
6	RnC	5.86
7	Resample	6.64
8	LDS	6.95
9	SqrtReWeight	7.23
10	ReWeight	7.82
11	CBLoss	8.14
12	DANN	11.00

Ranks are computed per dataset and averaged over all 4 datasets (lower is better). As shown in Table 19, `L-MDS` and `F-MDS` achieve the best average rank on the majority of all metrics: Overall, Attr Avg., Medium-Shot, Few-shot and Zero-shot.

Table 18: Average ranking of methods across metrics on CodeNet. Lower average rank is better. Rankings are computed over the 11 MAE metrics reported in the main table.

Rank	Method	Average Metric Rank
1	L-MDS	3.36
2	L-MDS + F-MDS	4.00
3	SqrtReWeight	4.55
4	F-MDS	5.18
4	CBLoss	5.18
4	Resample	5.18
7	ReWeight	6.55
8	LDS	6.95
9	DANN	8.00
10	ERM	8.36
11	RnC	9.73
12	GroupDRO	10.95

Table 19: Average ranking across datasets for different evaluation metrics. Lower average rank is better.

Method	Overall	Attr Avg.	Many Avg.	Medium Avg.	Few Avg.	Zero Avg.
L-MDS	**2.00**	**2.00**	4.50	4.00	**2.75**	4.00
F-MDS	2.75	2.75	5.50	**3.25**	3.25	**2.33**
SqrtReWeight	4.25	4.25	5.00	4.75	4.50	5.00
L-MDS + F-MDS	4.75	5.00	6.75	4.50	5.25	5.00
RnC	5.50	5.25	5.75	5.00	7.25	6.33
LDS	6.25	6.25	5.25	6.75	8.75	4.33
GroupDRO	7.50	7.25	8.50	5.00	6.75	7.67
ERM	7.50	7.50	**1.75**	6.25	9.00	7.33
Resample	7.50	7.50	9.00	8.75	4.75	7.00
CBLoss	9.25	9.25	8.75	9.50	7.00	8.00
ReWeight	10.25	10.25	10.25	10.00	7.00	9.67
DANN	10.50	10.50	7.00	10.25	11.50	11.33

B.2 Hyper-parameter choices for L-MDS and F-MDS

We explore the effects of different hyper-parameter choices on both L-MDS and F-MDS. Since we primarily use the Gaussian kernel for smoothing, we select different kernel sizes $k \in \{5, 9, 15\}$ and standard deviations $\sigma \in \{1, 2, 3\}$ for L-MDS. For F-MDS, we vary the choice of kernel size $k \in \{3, 5, 9\}$ and standard deviation $\sigma \in \{1, 2\}$. For F-MDS, we additionally experiment with two different reweighting methods: inverse reweight and square-root inverse reweight on our final weights.

UTKFace. We show results for UTKFace in Table 20. The overall performance for different hyper-parameter choices is quite stable. Interestingly, for UTKFace, L-MDS has better results with smaller standard deviations, while F-MDS has slightly better performance with the square-root inverse reweighting scheme.

SkyFinder. We report similar results for SkyFinder in Table 21, where both L-MDS and F-MDS demonstrate consistent performance metrics regardless of kernel size k or standard deviation σ. With larger kernel sizes of 9 and 15 in L-MDS, smaller standard deviations (e.g. $\sigma = 1$) generally lead to the best results, but gains are marginal. For F-MDS, we report that square-root inverse reweighting generally outperforms inverse reweighting in *many-shot* regions, while inverse reweighting yields slightly better results on *few-shot* and *zero-shot* regions.

PovertyMap. Again, minimal fluctuations in MAE and GM despite varying hyper-parameter choices reflect the robustness of L-MDS and F-MDS on the PovertyMap dataset. We report no consistent trend on kernel size and standard deviation for L-MDS- for instance, $k = 15$ and $\sigma = 1$ has

Table 20: Ablation study of different hyper-parameters for L-MDS and F-MDS on UTKFace

Method	k	σ	RW	Overall MAE↓	Overall GM↓	Attribute Avg↓	Attribute Worst↓	Many-shot Avg↓	Many-shot Worst↓	Medium-shot Avg↓	Medium-shot Worst↓	Few-shot Avg↓	Few-shot Worst↓	Zero-shot Avg↓	Zero-shot Worst↓
LDS + L-MDS variants															
L-MDS	5	1	Square-Root Inverse	7.34 ±0.1	3.97 ±0.0	7.21 ±0.1	9.03 ±0.2	4.58 ±0.1	17.12 ±2.3	6.06 ±0.1	17.85 ±1.1	6.87 ±0.1	13.04 ±0.5	9.88 ±0.1	70.04 ±13.7
		2	Square-Root Inverse	7.28 ±0.1	3.91 ±0.1	7.13 ±0.1	8.90 ±0.1	4.58 ±0.1	19.53 ±1.5	6.07 ±0.1	18.87 ±0.3	6.77 ±0.2	13.09 ±0.6	9.71 ±0.2	72.36 ±3.8
		3	Square-Root Inverse	7.41 ±0.1	4.05 ±0.1	7.27 ±0.1	8.96 ±0.1	4.66 ±0.2	18.67 ±1.9	6.15 ±0.1	18.15 ±0.4	6.77 ±0.1	12.35 ±0.4	9.96 ±0.3	77.14 ±9.5
	9	1	Square-Root Inverse	7.30 ±0.1	3.95 ±0.1	7.17 ±0.1	8.94 ±0.2	4.66 ±0.2	19.40 ±2.1	6.03 ±0.1	17.62 ±1.0	6.79 ±0.2	12.51 ±1.2	9.79 ±0.2	76.32 ±6.2
		2	Square-Root Inverse	7.36 ±0.2	3.98 ±0.0	7.22 ±0.2	8.92 ±0.3	4.57 ±0.1	18.97 ±1.3	6.13 ±0.1	17.90 ±1.5	6.87 ±0.1	12.44 ±0.9	9.86 ±0.4	73.51 ±8.0
		3	Square-Root Inverse	7.40 ±0.1	4.06 ±0.1	7.26 ±0.1	9.00 ±0.1	4.51 ±0.2	18.17 ±1.5	6.17 ±0.1	17.46 ±0.7	6.84 ±0.2	12.35 ±0.7	9.97 ±0.1	72.44 ±3.8
	15	1	Square-Root Inverse	7.27 ±0.1	3.90 ±0.1	7.14 ±0.1	8.94 ±0.2	4.58 ±0.1	16.84 ±2.6	6.09 ±0.1	17.58 ±1.0	6.75 ±0.1	12.17 ±0.7	9.69 ±0.3	78.96 ±5.6
		2	Square-Root Inverse	7.30 ±0.1	3.98 ±0.1	7.16 ±0.1	8.81 ±0.2	4.74 ±0.2	19.12 ±1.5	6.14 ±0.2	17.34 ±1.2	6.76 ±0.1	12.22 ±0.4	9.66 ±0.2	74.90 ±9.3
		3	Square-Root Inverse	7.38 ±0.1	3.98 ±0.1	7.24 ±0.1	8.91 ±0.2	4.62 ±0.1	18.26 ±0.6	6.16 ±0.1	18.94 ±1.9	6.88 ±0.2	12.67 ±1.2	9.86 ±0.3	77.52 ±5.3
LDS + F-MDS variants															
F-MDS	3	1	Inverse	7.36 ±0.1	4.02 ±0.1	7.22 ±0.1	8.97 ±0.1	4.90 ±0.2	18.65 ±1.7	6.08 ±0.1	17.66 ±0.8	6.58 ±0.2	12.41 ±0.6	9.86 ±0.2	77.05 ±7.5
			Square-Root Inverse	7.40 ±0.1	3.98 ±0.1	7.26 ±0.1	9.03 ±0.2	4.47 ±0.1	18.20 ±1.3	6.16 ±0.1	18.47 ±0.5	6.90 ±0.2	12.34 ±0.6	9.98 ±0.2	72.79 ±12.7
		2	Inverse	7.46 ±0.1	4.05 ±0.0	7.31 ±0.1	9.05 ±0.1	4.99 ±0.1	18.59 ±1.9	6.09 ±0.1	17.97 ±0.7	6.65 ±0.1	13.11 ±0.9	10.06 ±0.1	81.63 ±5.1
			Square-Root Inverse	7.31 ±0.1	3.90 ±0.1	7.17 ±0.1	8.89 ±0.2	4.58 ±0.1	17.97 ±0.9	6.10 ±0.0	17.60 ±0.7	6.76 ±0.2	12.82 ±0.8	9.77 ±0.2	75.33 ±14.5
	5	1	Inverse	7.45 ±0.1	4.04 ±0.1	7.30 ±0.1	9.02 ±0.2	4.97 ±0.1	19.39 ±1.9	6.11 ±0.1	17.26 ±1.1	6.69 ±0.1	13.10 ±1.6	10.00 ±0.2	81.48 ±5.6
			Square-Root Inverse	7.43 ±0.1	3.99 ±0.1	7.30 ±0.1	9.17 ±0.2	4.57 ±0.1	17.96 ±2.3	6.09 ±0.1	17.85 ±1.0	6.96 ±0.1	12.55 ±0.7	10.06 ±0.3	77.69 ±7.7
		2	Inverse	7.46 ±0.1	4.05 ±0.0	7.31 ±0.1	9.11 ±0.0	4.92 ±0.1	17.63 ±1.8	6.13 ±0.1	17.02 ±0.7	6.55 ±0.1	12.25 ±0.5	10.08 ±0.1	82.71 ±3.6
			Square-Root Inverse	7.32 ±0.1	4.00 ±0.1	7.18 ±0.1	8.93 ±0.2	4.55 ±0.1	18.94 ±1.1	6.11 ±0.1	17.98 ±0.9	6.80 ±0.1	12.45 ±0.5	9.79 ±0.2	72.24 ±12.0
	9	1	Inverse	7.42 ±0.1	4.02 ±0.1	7.28 ±0.1	8.96 ±0.1	4.97 ±0.1	19.75 ±2.3	6.09 ±0.1	17.24 ±1.5	6.67 ±0.2	13.30 ±1.4	9.97 ±0.2	77.73 ±6.7
			Square-Root Inverse	7.22 ±0.1	3.91 ±0.1	7.08 ±0.1	8.71 ±0.2	4.65 ±0.2	17.49 ±1.9	6.08 ±0.1	17.91 ±0.2	6.71 ±0.2	12.43 ±1.4	9.54 ±0.4	68.81 ±8.1
		2	Inverse	7.36 ±0.1	4.00 ±0.0	7.21 ±0.1	9.08 ±0.0	4.88 ±0.1	18.43 ±2.2	6.03 ±0.0	17.35 ±0.9	6.59 ±0.1	12.93 ±0.6	9.92 ±0.2	74.21 ±4.2
			Square-Root Inverse	7.39 ±0.1	4.02 ±0.1	7.26 ±0.1	9.00 ±0.2	4.67 ±0.1	17.68 ±1.6	6.07 ±0.1	17.80 ±0.9	6.85 ±0.2	13.29 ±1.3	9.97 ±0.2	71.28 ±10.1

Table 21: Ablation study of different hyper-parameters for L-MDS and F-MDS on SkyFinder

Method	k	σ	RW	Overall MAE↓	Overall GM↓	Attribute Avg↓	Attribute Worst↓	Many-shot Avg↓	Many-shot Worst↓	Medium-shot Avg↓	Medium-shot Worst↓	Few-shot Avg↓	Few-shot Worst↓	Zero-shot Avg↓	Zero-shot Worst↓
LDS + L-MDS variants															
L-MDS	5	1	Square-Root Inverse	3.54 ±0.0	2.17 ±0.0	3.27 ±0.0	5.94 ±0.1	2.35 ±0.0	6.88 ±0.5	2.94 ±0.0	12.40 ±1.8	4.19 ±0.0	23.21 ±1.2	4.80 ±0.0	30.94 ±1.5
		2	Square-Root Inverse	3.57 ±0.0	2.20 ±0.0	3.30 ±0.0	5.82 ±0.1	2.39 ±0.0	6.71 ±0.7	2.98 ±0.0	13.13 ±1.5	4.24 ±0.0	23.20 ±0.8	4.79 ±0.1	33.25 ±2.9
		3	Square-Root Inverse	3.59 ±0.0	2.22 ±0.0	3.30 ±0.0	5.80 ±0.1	2.33 ±0.0	6.14 ±0.6	2.99 ±0.0	11.33 ±0.3	4.26 ±0.1	23.38 ±0.5	4.83 ±0.0	28.82 ±2.7
	9	1	Square-Root Inverse	3.57 ±0.0	2.21 ±0.0	3.30 ±0.0	5.89 ±0.2	2.29 ±0.0	6.25 ±1.1	3.00 ±0.0	12.48 ±1.2	4.21 ±0.1	23.51 ±1.0	4.82 ±0.1	30.54 ±2.3
		2	Square-Root Inverse	3.55 ±0.0	2.18 ±0.0	3.28 ±0.0	5.79 ±0.1	2.35 ±0.0	6.38 ±0.8	2.95 ±0.0	12.09 ±1.1	4.22 ±0.0	23.74 ±1.5	4.77 ±0.1	28.15 ±3.0
		3	Square-Root Inverse	3.54 ±0.0	2.19 ±0.0	3.27 ±0.0	5.81 ±0.2	2.38 ±0.0	6.86 ±0.6	2.95 ±0.0	11.63 ±1.0	4.17 ±0.0	23.63 ±0.7	4.78 ±0.0	31.47 ±2.3
	15	2	Square-Root Inverse	3.57 ±0.0	2.19 ±0.0	3.30 ±0.0	5.83 ±0.1	2.34 ±0.0	6.75 ±0.4	2.98 ±0.0	13.34 ±1.1	4.24 ±0.1	23.34 ±1.9	4.77 ±0.1	33.20 ±4.4
		3	Square-Root Inverse	3.54 ±0.0	2.17 ±0.0	3.26 ±0.0	5.83 ±0.1	2.32 ±0.1	6.48 ±0.6	2.95 ±0.0	11.59 ±0.6	4.20 ±0.0	22.27 ±1.7	4.74 ±0.0	29.98 ±2.6
LDS + F-MDS variants															
F-MDS	3	1	Inverse	3.62 ±0.0	2.24 ±0.0	3.32 ±0.0	5.74 ±0.1	2.68 ±0.1	7.18 ±0.4	3.10 ±0.0	12.84 ±0.5	4.13 ±0.0	20.03 ±0.8	4.79 ±0.0	30.30 ±3.2
			Square-Root Inverse	3.56 ±0.0	2.19 ±0.0	3.28 ±0.0	5.76 ±0.1	2.41 ±0.1	7.27 ±0.3	2.97 ±0.0	11.76 ±1.0	4.20 ±0.0	23.03 ±0.6	4.78 ±0.0	30.44 ±2.0
		2	Inverse	3.63 ±0.0	2.24 ±0.0	3.33 ±0.0	5.88 ±0.1	2.62 ±0.1	7.50 ±0.2	3.10 ±0.0	13.07 ±1.1	4.17 ±0.1	21.27 ±1.4	4.76 ±0.0	28.81 ±1.6
			Square-Root Inverse	3.59 ±0.0	2.20 ±0.0	3.30 ±0.0	5.80 ±0.0	2.42 ±0.1	6.79 ±1.0	3.00 ±0.0	12.91 ±1.0	4.23 ±0.1	23.46 ±0.7	4.83 ±0.1	29.68 ±1.9
	5	1	Inverse	3.57 ±0.0	2.21 ±0.0	3.28 ±0.0	5.79 ±0.1	2.55 ±0.0	6.65 ±0.5	3.05 ±0.0	12.73 ±1.0	4.12 ±0.1	21.91 ±1.3	4.72 ±0.0	29.46 ±2.8
			Square-Root Inverse	3.57 ±0.0	2.19 ±0.0	3.29 ±0.0	5.83 ±0.1	2.31 ±0.1	6.41 ±0.5	2.98 ±0.0	12.18 ±1.3	4.23 ±0.0	22.98 ±0.9	4.80 ±0.0	31.36 ±1.6
		2	Inverse	3.58 ±0.0	2.21 ±0.0	3.29 ±0.0	5.76 ±0.2	2.51 ±0.0	6.87 ±0.5	3.04 ±0.0	11.72 ±0.8	4.13 ±0.1	20.41 ±1.3	4.78 ±0.1	31.76 ±2.7
			Square-Root Inverse	3.56 ±0.0	2.18 ±0.0	3.28 ±0.0	5.88 ±0.2	2.30 ±0.0	6.17 ±0.6	2.95 ±0.0	12.58 ±0.2	4.25 ±0.0	22.52 ±0.7	4.80 ±0.1	32.55 ±2.8
	9	1	Inverse	3.59 ±0.0	2.22 ±0.0	3.30 ±0.0	5.84 ±0.1	2.56 ±0.1	6.89 ±0.7	3.07 ±0.0	12.67 ±0.6	4.12 ±0.0	19.94 ±2.2	4.72 ±0.1	29.46 ±1.8
			Square-Root Inverse	3.58 ±0.0	2.20 ±0.0	3.30 ±0.0	5.97 ±0.1	2.33 ±0.1	6.48 ±0.5	2.97 ±0.0	12.45 ±1.9	4.27 ±0.0	23.02 ±0.5	4.79 ±0.1	30.35 ±1.4
		2	Inverse	3.58 ±0.0	2.20 ±0.0	3.29 ±0.0	5.76 ±0.1	2.52 ±0.1	6.98 ±0.6	3.04 ±0.0	13.62 ±0.6	4.13 ±0.1	21.86 ±1.3	4.76 ±0.0	28.52 ±1.2
			Square-Root Inverse	3.56 ±0.0	2.18 ±0.0	3.29 ±0.0	5.81 ±0.2	2.33 ±0.1	6.44 ±0.3	2.97 ±0.0	11.86 ±0.4	4.22 ±0.0	21.40 ±1.1	4.74 ±0.0	30.47 ±2.0

the lowest overall MAE and GM, while $k = 9$ and $\sigma = 1$ performs the worse, and $k = 5$ and $\sigma = 1$ is in-between the two. Reweighting methods balance out overall for F-MDS, as square-root inverse reweighting performs better in *many-shot* regions overall, while inverse reweighting does better on more data-scarce regions.

B.3 Kernel Type for L-MDS and F-MDS

We further investigate the impact of different kernel choices for L-MDS and F-MDS, beyond the default configuration that uses Gaussian kernels. We experiment with three different kernel types: *Gaussian*, *Laplacian*, and *Triangular* kernel, evaluate their effects on L-MDS and F-MDS. We use kernel size $l = 5$ and the standard deviation $\sigma = 2$ for all kernels and report results on PovertyMap in Table 23. As the table illustrates, all kernel types provide improvements over the ERM baseline, especially in *few-shot* and *zero-shot* regions. Moreover, Laplacian gives the best results for both L-MDS and F-MDS. These results suggest that both L-MDS and F-MDS are robust to different smoothing kernel types.

B.4 Training Loss for L-MDS and F-MDS

In the main paper, we use L_1 loss during training for all datasets. Besides L_1, we also study the effect of different training loss functions on L-MDS and F-MDS. Specifically, we compare three common loss functions that people use for regression tasks: L_1 loss, MSE loss, and the Huber loss. Results on PovertyMap are shown in Table 24. We notice that there are no significant performance differences

Table 22: Ablation study of different hyper-parameters for L-MDS and F-MDS on PovertyMap

Method	k	σ	RW	Overall MAE↓	Overall GM↓	Attribute Avg↓	Attribute Worst↓	Many-shot Avg↓	Many-shot Worst↓	Medium-shot Avg↓	Medium-shot Worst↓	Few-shot Avg↓	Few-shot Worst↓	Zero-shot Avg↓	Zero-shot Worst↓
LDS + L-MDS variants															
L-MDS	5	1	Square-Root Inverse	0.492 ±0.0	0.320 ±0.0	0.489 ±0.0	0.659 ±0.0	0.321 ±0.1	0.716 ±0.2	0.338 ±0.0	1.383 ±0.2	0.478 ±0.0	2.278 ±0.2	0.717 ±0.0	1.992 ±0.0
		2	Square-Root Inverse	0.494 ±0.0	0.319 ±0.0	0.492 ±0.0	0.694 ±0.0	0.306 ±0.0	0.626 ±0.1	0.348 ±0.0	1.338 ±0.1	0.471 ±0.0	2.369 ±0.1	0.728 ±0.0	1.992 ±0.1
		3	Square-Root Inverse	0.492 ±0.0	0.320 ±0.0	0.490 ±0.0	0.657 ±0.0	0.324 ±0.0	0.824 ±0.1	0.331 ±0.0	1.289 ±0.0	0.477 ±0.0	2.354 ±0.1	0.728 ±0.0	1.911 ±0.1
	9	1	Square-Root Inverse	0.496 ±0.0	0.327 ±0.0	0.493 ±0.0	0.662 ±0.0	0.320 ±0.1	0.666 ±0.2	0.347 ±0.0	1.307 ±0.1	0.475 ±0.0	2.302 ±0.2	0.729 ±0.0	2.035 ±0.1
		2	Square-Root Inverse	0.486 ±0.0	0.317 ±0.0	0.484 ±0.0	0.666 ±0.0	0.271 ±0.0	0.535 ±0.2	0.336 ±0.0	1.417 ±0.1	0.467 ±0.0	2.385 ±0.1	0.720 ±0.0	1.987 ±0.1
		3	Square-Root Inverse	0.494 ±0.0	0.318 ±0.0	0.492 ±0.0	0.656 ±0.0	0.310 ±0.0	0.683 ±0.1	0.343 ±0.0	1.321 ±0.1	0.476 ±0.0	2.265 ±0.1	0.727 ±0.0	2.045 ±0.1
	15	1	Square-Root Inverse	0.489 ±0.0	0.313 ±0.0	0.486 ±0.0	0.665 ±0.0	0.300 ±0.1	0.643 ±0.2	0.340 ±0.0	1.437 ±0.2	0.470 ±0.0	2.228 ±0.1	0.717 ±0.0	2.082 ±0.1
		2	Square-Root Inverse	0.493 ±0.0	0.324 ±0.0	0.491 ±0.0	0.654 ±0.0	0.317 ±0.1	0.681 ±0.3	0.337 ±0.0	1.370 ±0.1	0.479 ±0.0	2.309 ±0.2	0.724 ±0.0	2.051 ±0.1
		3	Square-Root Inverse	0.495 ±0.0	0.321 ±0.0	0.492 ±0.0	0.655 ±0.0	0.314 ±0.1	0.669 ±0.3	0.347 ±0.0	1.306 ±0.1	0.471 ±0.0	2.241 ±0.1	0.733 ±0.0	2.026 ±0.0
LDS + F-MDS variants															
F-MDS	3	1	Inverse	0.497 ±0.0	0.319 ±0.0	0.494 ±0.0	0.670 ±0.0	0.361 ±0.1	0.751 ±0.1	0.344 ±0.0	1.359 ±0.1	0.465 ±0.0	2.170 ±0.1	0.756 ±0.0	1.994 ±0.1
			Square-Root Inverse	0.498 ±0.0	0.323 ±0.0	0.495 ±0.0	0.675 ±0.0	0.316 ±0.1	0.637 ±0.2	0.353 ±0.0	1.348 ±0.1	0.475 ±0.0	2.325 ±0.1	0.729 ±0.0	2.031 ±0.1
		2	Inverse	0.490 ±0.0	0.314 ±0.0	0.488 ±0.0	0.673 ±0.0	0.330 ±0.0	0.653 ±0.1	0.343 ±0.0	1.347 ±0.1	0.467 ±0.0	2.227 ±0.1	0.726 ±0.0	2.046 ±0.1
			Square-Root Inverse	0.488 ±0.0	0.319 ±0.0	0.485 ±0.0	0.641 ±0.0	0.274 ±0.0	0.545 ±0.1	0.323 ±0.0	1.269 ±0.1	0.476 ±0.0	2.246 ±0.1	0.725 ±0.0	1.949 ±0.1
	5	1	Inverse	0.493 ±0.0	0.320 ±0.0	0.490 ±0.0	0.663 ±0.0	0.369 ±0.1	0.759 ±0.3	0.347 ±0.0	1.328 ±0.2	0.467 ±0.0	2.264 ±0.1	0.732 ±0.0	2.000 ±0.1
			Square-Root Inverse	0.490 ±0.0	0.316 ±0.0	0.487 ±0.0	0.651 ±0.0	0.295 ±0.1	0.624 ±0.2	0.333 ±0.0	1.272 ±0.1	0.474 ±0.0	2.229 ±0.2	0.723 ±0.0	1.997 ±0.1
		2	Inverse	0.493 ±0.0	0.321 ±0.0	0.490 ±0.0	0.650 ±0.0	0.357 ±0.1	0.695 ±0.2	0.342 ±0.0	1.304 ±0.1	0.471 ±0.0	2.224 ±0.1	0.728 ±0.0	2.000 ±0.1
			Square-Root Inverse	0.489 ±0.0	0.315 ±0.0	0.486 ±0.0	0.655 ±0.0	0.262 ±0.0	0.538 ±0.1	0.340 ±0.0	1.318 ±0.1	0.476 ±0.0	2.414 ±0.1	0.707 ±0.0	2.088 ±0.1
	9	1	Inverse	0.500 ±0.0	0.325 ±0.0	0.497 ±0.0	0.670 ±0.0	0.335 ±0.0	0.708 ±0.1	0.346 ±0.0	1.352 ±0.1	0.471 ±0.0	2.205 ±0.1	0.753 ±0.0	1.950 ±0.1
			Square-Root Inverse	0.489 ±0.0	0.315 ±0.0	0.487 ±0.0	0.665 ±0.0	0.289 ±0.0	0.603 ±0.1	0.338 ±0.0	1.398 ±0.1	0.468 ±0.0	2.381 ±0.1	0.728 ±0.0	2.067 ±0.0
		2	Inverse	0.494 ±0.0	0.317 ±0.0	0.491 ±0.0	0.664 ±0.0	0.328 ±0.0	0.673 ±0.1	0.340 ±0.0	1.344 ±0.1	0.470 ±0.0	2.326 ±0.1	0.736 ±0.0	2.042 ±0.1
			Square-Root Inverse	0.488 ±0.0	0.318 ±0.0	0.485 ±0.0	0.670 ±0.0	0.278 ±0.0	0.554 ±0.1	0.327 ±0.0	1.307 ±0.1	0.477 ±0.0	2.492 ±0.1	0.719 ±0.0	2.057 ±0.0

Table 23: Ablation study of different kernel types for L-MDS and F-MDS on PovertyMap

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
ERM	0.504	0.502	0.679	0.256	0.504	0.335	1.356	0.494	2.452	0.744	1.996
L-MDS:											
GAUSSIAN KERNEL	0.486	0.484	0.666	0.271	0.535	0.336	1.417	0.467	2.385	0.720	1.987
LAPLACIAN KERNEL	0.479	0.477	0.624	0.243	0.534	0.312	1.191	0.468	2.494	0.716	2.037
TRIANGULAR KERNEL	0.480	0.478	0.626	0.221	0.359	0.323	1.230	0.481	2.302	0.684	2.057
F-MDS:											
GAUSSIAN KERNEL	0.488	0.485	0.670	0.278	0.554	0.327	1.307	0.477	2.492	0.719	2.057
LAPLACIAN KERNEL	0.485	0.482	0.639	0.434	0.862	0.356	1.325	0.452	2.084	0.713	1.929
TRIANGULAR KERNEL	0.486	0.483	0.668	0.371	0.558	0.335	1.277	0.459	2.541	0.733	2.007

Table 24: Ablation study of different loss functions used during training for L-MDS and F-MDS on PovertyMap

Algorithm	Overall	Test Error (by attribute)		Test Error (by shot)							
		Average	Worst	Many		Medium		Few		Zero	
				Average	Worst	Average	Worst	Average	Worst	Average	Worst
ERM	0.504	0.502	0.679	0.256	0.504	0.335	1.356	0.494	2.452	0.744	1.996
L-MDS:											
L1	0.486	0.484	0.666	0.271	0.535	0.336	1.417	0.467	2.385	0.720	1.987
MSE	0.481	0.479	0.700	0.321	0.749	0.307	1.227	0.465	2.501	0.734	2.142
HUBER LOSS	0.484	0.481	0.650	0.341	0.806	0.335	1.429	0.471	2.142	0.699	1.981
F-MDS:											
L1	0.488	0.485	0.670	0.278	0.554	0.327	1.307	0.477	2.492	0.719	2.057
L2	0.494	0.490	0.665	0.160	0.199	0.316	1.285	0.475	2.615	0.765	2.287
HUBER LOSS	0.486	0.483	0.667	0.285	0.378	0.335	1.279	0.471	2.422	0.713	1.967

between the losses and all three losses gain improvements from the baseline, indicating that L-MDS and F-MDS are robust to the choice of different loss functions.

Table 25: Interpolation & extrapolation results on a curated subset of UTKFace

Metrics	MAE ↓				GM ↓			
Shot	All	w/ data	Interp.	Extrap.	All	w/ data	Interp.	Extrap.
ERM	14.02	10.10	13.53	25.71	8.20	5.79	9.42	17.92
F-MDS	12.65	10.05	11.98	20.89	7.56	5.93	7.86	14.07
L-MDS	13.11	10.30	12.95	21.21	7.97	6.20	8.75	14.08
Ours (best) vs. ERM	**+1.37**	**+0.05**	**+1.56**	**+4.82**	**+0.65**	**-0.15**	**+1.56**	**+3.84**

Table 26: Results on UTKFace with 50% training data

Algorithm	Overall		MAE (by attribute)		MAE (by shot)							
	MAE	GM	Average	Worst	Many		Medium		Few		Zero	
					Average	Worst	Average	Worst	Average	Worst	Average	Worst
ERM	8.13	7.84	7.98	10.33	2.83	6.42	6.06	23.12	7.72	17.65	10.99	82.25
L-MDS	7.99	7.74	7.84	9.71	3.22	5.97	6.32	19.85	7.55	14.77	10.41	82.99
F-MDS	8.02	7.74	7.86	9.49	3.22	5.99	6.31	18.09	7.50	15.17	10.52	82.91
Ours (best) vs. ERM	**+0.14**	**+0.10**	**+0.14**	**+0.84**	**-0.39**	**+0.45**	**-0.25**	**+5.03**	**+0.22**	**+2.88**	**+0.58**	**-0.66**

Table 27: Results on UTKFace with 20% training data

Algorithm	Overall		MAE (by attribute)		MAE (by shot)							
	MAE	GM	Average	Worst	Many		Medium		Few		Zero	
					Average	Worst	Average	Worst	Average	Worst	Average	Worst
ERM	10.33	9.96	10.11	12.14	3.75	4.02	6.49	24.89	9.20	22.69	13.69	95.90
L-MDS	10.08	9.79	9.89	11.19	5.17	5.61	6.74	23.44	8.87	22.67	13.14	82.98
F-MDS	10.12	9.81	9.92	11.66	4.60	5.73	6.77	20.27	8.91	19.17	13.18	69.23
Ours (best) vs. ERM	**+0.25**	**+0.17**	**+0.22**	**+0.95**	**-0.85**	**-1.59**	**-0.25**	**+4.62**	**+0.33**	**+3.52**	**+0.55**	**+26.67**

B.5 Analysis of Interpolation & Extrapolation

We construct a curated subset of UTKFace with missing target regions in the train sets while evaluating on the original test set. As shown in Table 25, both L-MDS and F-MDS consistently improve over ERM across both MAE and GM metrics. Notably, the gains are more pronounced in the *interpolation* and *extrapolation* regions, suggesting that our smoothing methods help transfer information across related target regions and improve generalization to unseen or underrepresented targets.

B.6 Resilience to Reduced Training Data

The success of modern deep learning methods has largely relied on the availability of large-scale labeled datasets. However, collecting and annotating such datasets is often costly and time-consuming in real-world applications. Thus, it is important to evaluate models under limited training data. We subsample UTKFace to 50%, 20%, and 10% of the original training data, and train ERM, L-MDS, and F-MDS separately on each subsampled subset. Shot-wise results for 50%, 20% and 10% are in Table 26, Table 27, Table 28. We observe L-MDS and F-MDS are more robust to reduced training data and achieve better performance gains. Moreover, as the training set becomes smaller, the performance gap between our methods and the baseline increases and both L-MDS and F-MDS gain from *zero-shot* and *few-shot* regions.

B.7 Broader Impacts

We believe L-MDS and F-MDS can have positive societal impact by making continuous prediction systems more reliable in real-world settings where data is observational, imbalanced, and affected

Table 28: Results on UTKFace with 10% training data

Algorithm	Overall		MAE (by attribute)		MAE (by shot)							
	MAE	GM	Average	Worst	Many		Medium		Few		Zero	
					Average	Worst	Average	Worst	Average	Worst	Average	Worst
ERM	14.02	13.42	13.65	15.81	−−	−−	8.32	31.49	10.83	34.13	18.55	86.31
L-MDS	13.11	12.64	12.80	14.75	−−	−−	9.28	34.90	10.71	27.64	16.35	85.21
F-MDS	12.65	12.19	12.35	14.19	−−	−−	9.04	35.48	10.46	26.35	15.65	85.07
Ours (best) vs. ERM	**+1.37**	**+1.23**	**+1.30**	**+1.62**	−−	−−	**-0.72**	**-3.99**	**+0.37**	**+7.78**	**+2.90**	**+1.24**

Table 29: Overview of the four DSR datasets used in our experiments.

Dataset	# Attrs	Target Range		Attr Density		Split Sizes		
		Min	Max	Min	Max	Train	Val	Test
UTKFace	5	1	116	830	8,392	17,620	2,753	3,730
SkyFinder	47	-27.2	50.0	14	3,464	64,945	9,335	6,766
PovertyMap	20	-1.1	2.5	97	1,159	6,034	475	545
CodeNet	13	0	1,000	1,500	1,500	19,500	6,374	6,374

by deployment shifts. Such settings include medical risk prediction, environmental sensing, poverty estimation, and other domains where models may otherwise rely on shortcuts tied to demographics, locations, devices, or data-collection conditions. By explicitly evaluating subgroup errors and improving performance in sparse or unseen target regions, our work may help reduce hidden failure modes missed by average regression metrics.

At the same time, several risks require careful consideration. First, our methods depend on the availability and quality of spurious attribute annotations. If the attributes are incomplete, noisy, or themselves sensitive, the resulting model may provide a false sense of robustness or introduce new biases. Second, improving regression robustness can make it easier to deploy models for predicting sensitive personal attributes, such as health or socioeconomic status, from human data. Such uses may reinforce discrimination if the predictions are used for screening, ranking, surveillance, or resource allocation without proper oversight. Finally, better interpolation and extrapolation over continuous targets may be misused to justify predictions in regions with little or no reliable training evidence. Therefore, our methods should be used with careful validation, transparent reporting of subgroup performance, and domain-specific ethical review before deployment in high-stakes applications.

C Dataset Details

We provide detailed information of the four real-world datasets and one synthetic dataset used in our experiments to investigate DSR in this section. Table 29 provides an overview of the each dataset.

C.1 ColoredRotatedMNIST

Dataset Construction. ColoredRotatedMNIST is a synthetic regression dataset built on MNIST (LeCun et al., 1998) digit "2". Each image is rotated to a continuous angle $y \in (0°, 180°)$ and placed on a solid-color background serving as the spurious attribute $a \in \{\text{Red}, \text{Blue}, \text{Green}, \text{Yellow}\}$. The task is to predict the rotation angle y from the image. In the training set, we divide the $180°$ range into four equal subintervals of $45°$ each, with each color assigned a dominant subinterval: Red to $[0°, 45°)$, Blue to $[45°, 90°)$, Green to $[90°, 135°)$, and Yellow to $[135°, 180°)$. Within its dominant subinterval, each color has 10 samples per degree (450 samples total); outside its dominant subinterval, each color has only 50 samples uniformly drawn from each of the remaining three $45°$ subintervals (150 samples total). This induces a strong spurious correlation between background color and rotation angle in training. The validation and test sets are uniformly distributed across all (y, a) combinations (10 samples per degree per color), providing an unbiased evaluation of generalization.

Table 30: Results on `ColoredRotatedMNIST` using ERM across MAE, MSE, and GM metrics

Metric	Test Error (by attribute)		Test Error (by shot)							
	Average	Worst	Many		Medium		Few		Zero	
			Average	Worst	Average	Worst	Average	Worst	Average	Worst
MAE	12.90	14.16	7.47	21.27	20.18	20.81	15.26	91.90	13.50	74.29
MSE	509.77	671.32	122.76	1335.52	819.46	908.84	700.00	12399.98	507.52	9507.13
GM	6.88	7.18	4.52	11.23	13.12	13.45	8.05	54.82	7.56	37.89

Detailed ERM Results Table 30 reports results on `ColoredRotatedMNIST` trained using ERM to evaluate the impact of DSR across MAE, MSE, and GM metrics. Many-shot groups achieve the lowest average errors across all metrics, confirming that sufficient training coverage leads to better generalization. In contrast, few- and zero-shot groups suffer the most, with worst-case errors far exceeding their averages, highlighting the existence of DSR.

C.2 UTKFace

The `UTKFace` dataset (Zhang et al., 2017) is a collection of more than 20,000 facial images, each labeled with age, gender, ethnicity. We predict age as the regression task, and we employ the five ethnicity groups as the spurious attribute defined as White, Black, Asian, Indian, and Others (e.g. Hispanic, Latino, Middle Eastern). From the complete dataset, we construct a training set with 17,620 images, a validation set with 2,753 images, and a test set with 3,730 images. Age ranges from 1 to 116 years. The train set exhibits significant imbalance across ethnic groups, with the White ethnicity comprising 10,222 samples (42.4%) while the Others ethnicity accounts for only 1,711 samples (7.1%). The test and validation sets are constructed uniformly.

C.3 SkyFinder

The original `SkyFinder` dataset (Mihail et al., 2016) dataset is a large-scale dataset of pixel-annotated images of the sky and other regions taken by outdoor webcam images. We use the camera ID as the spurious attribute and the in-the-wild temperature associated with each images as the regression target. We stratify the data across 47 webcams and split the full dataset into 64,945 training images, 9,335 validation images, and 6,766 test images. Temperature ranges from -27.2 to 50.0 degrees Celsius. The train set exhibits significant imbalance across cameras, with the most frequent camera (ID 684) comprising 3,464 samples (5.3%) while the least frequent camera (ID 4232) accounts for only 14 samples (0.02%). The test and validation sets are constructed more uniformly.

C.4 PovertyMap

The `PovertyMap` dataset is a subset of the original PovertyMap-WILDS (Koh et al., 2021) benchmark dataset. The dataset consists of satellite images of rural and urban regions across multiple countries. We consider the spurious attribute as the country an image is of, and we predict on poverty index as the regression target. We create a training set of 6,034 images total, with 475 validation images and 545 test images. The wealth index ranges from -1.1 to 2.5. The train set exhibits significant imbalance across countries, with Tanzania comprising 1,521 samples (25.2%) while Togo accounts for only 97 samples (1.6%), reflecting the uneven country representation in the original survey data. The test and validation sets are constructed to have a more uniform distribution across countries.

C.5 CodeNet

The original `CodeNet` dataset (Puri et al., 2021) contains 14M coding samples along with metadata on metrics like programming language, code size, and code execution time. The pre-trained RLM-GemmaS-Code-V0 model filters the dataset to 7.3M "Accepted" solutions (Akhauri et al., 2025) and predicts over the memory column. To avoid data leakage, we predict the CPU execution time column as the regression target instead, and we clamp target values between 0 and 1000. We consider

programming language as the spurious attribute. Although the original dataset includes samples from 57 different programming languages, a select few languages (Python, C++, Java) dominate, while other languages have few samples. We curate a subset of the filtered dataset by taking samples from the 13 most common programming languages with at least 10K samples each within the target range. We then create a training set with 19,500 total samples, where each programming language has 1,500 samples and the target distribution follows a Gaussian distribution. We make the length of each bin 20 to construct the normal distribution. From the filtered dataset, we also curate the test and validation sets with approximately 500 samples each per language for a total of 6,374 samples per set and 12,748 total. Both the test and validation sets follow a uniform distribution per language with a bin size of 50.

D Experimental Settings

Network Architecture and Experiment Settings. We use ResNet-18 (He et al., 2016) for `UTKFace`, `SkyFinder`, and `PovertyMap`, and RLM-GemmaS-Code-V0 (Akhauri et al., 2025) for `CodeNet`.

Image Regression Datasets. All experiments on image regression datasets (`UTKFace`, `SkyFinder`, `PovertyMap`) are trained using a single NVIDIA A40 GPU with batch size 256 for 400 epochs. We use SGD with learning rate 0.2, momentum 0.9, weight decay 10^{-4}, and a cosine annealing learning rate schedule with decay rate 0.1 applied over 400 epochs.

For DANN (Ganin et al., 2016), a 3-layer MLP domain classifier (hidden width 256) is attached to the shared encoder via a gradient reversal layer, with encoder and discriminator learning rates both set to 0.2, adversarial weight $\lambda = 1.0$, and 1 discriminator step per generator step.

For GroupDRO (Sagawa et al., 2020), we use the same base optimizer as above with group step size $\eta = 0.01$.

For RnC (Zha et al., 2023), training follows a two-phase procedure. In the first phase, the encoder is trained for 400 epochs with learning rate 0.5, momentum 0.9, weight decay 10^{-4}, temperature $\tau = 2$, L_1 label distance, and L_2 feature similarity. In the second phase, the encoder is frozen and a linear regressor is trained on top for 100 epochs using SGD with learning rate 0.05, momentum 0.9, weight decay 0, and cosine annealing schedule with decay rate 0.2.

For both `L-MDS` and `F-MDS`, we use kernel size $k = 5$ and standard deviation $\sigma = 2$ with `sqrt_inv` reweighting. For `F-MDS` specifically, the feature centroid update begins at epoch 5.

Code Regression Dataset. Experiments on the code regression dataset fine-tune a pretrained seq2seq LLM (Akhauri et al., 2025) (`akhauriyash/RLM-GemmaS-Code-v0`) on a single NVIDIA A40 GPU, where only the decoder is fine-tuned while the encoder is kept frozen. We use AdamW with a learning rate of 2e-5, weight decay of 0.01, gradient clipping norm of 1.0, batch size of 16, and train for 20 epochs. For both `L-MDS` and `F-MDS`, we use kernel size $k = 5$ and standard deviation $\sigma = 2$ with `sqrt_inv` reweighting.