Multi-group Learning for Hierarchical Groups

Samuel Deng[1]   **Daniel Hsu**[1]

Abstract

The multi-group learning model formalizes the learning scenario in which a single predictor must generalize well on multiple, possibly overlapping subgroups of interest. We extend the study of multi-group learning to the natural case where the groups are hierarchically structured. We design an algorithm for this setting that outputs an interpretable and deterministic decision tree predictor with near-optimal sample complexity. We then conduct an empirical evaluation of our algorithm and find that it achieves attractive generalization properties on real datasets with hierarchical group structure.

1. Introduction

In the classical statistical learning setup, the goal is to construct a predictor with high *average* accuracy. However, in many practical learning scenarios, an aggregate, on-average measure of performance is insufficient. In general, average-case performance can obscure performance for subgroups of examples — a predictor that boasts 95% accuracy on average might only be 50% accurate on an important subgroup comprising 10% of the population. Such a subgroup might be difficult for a predictor trained for aggregate performance because it is not represented well during training (Oakden-Rayner et al., 2020) or it admits spurious correlations (Borkan et al., 2019).

Recent work has shown that, in various learning domains, constructing a predictor that performs well on multiple subgroups is crucial. For instance, when *fairness* is a concern, a natural desideratum is that a predictor be accurate not only on average, but also conditional on possibly intersecting subgroups such as race and gender (Hardt et al., 2016; Diana et al., 2021). In medical imaging, a model might sys-

tematically err on rarer cancers to achieve a higher average accuracy, leading to possibly life-threatening predictions on individuals with the rare cancer (Oakden-Rayner et al., 2020). Similar demands for group-wise accuracy appear in domains as varied as medical imaging (Bissoto et al., 2019; Oakden-Rayner et al., 2020; DeGrave et al., 2021), facial recognition (Buolamwini & Gebru, 2018; Kuehlkamp et al., 2017), object recognition (De Vries et al., 2019), and NLP (Orr et al., 2020; Varma et al., 2021; Borkan et al., 2019). A common theme is that these high error subgroups come from some *hierarchical group structure*. Demographic subgroups in fairness-aware scenarios naturally arrange via, say, race, gender, and age (Borkan et al., 2019) or subgroups of face images in facial recognition naturally arrange via, say, gender, facial expression, and hair color (Liu et al., 2015).

Motivated by these concerns, we study the *multi-group (agnostic PAC) learning* model first proposed by Rothblum & Yona (2021). In multi-group learning, there is a collection of *groups* $\mathcal{G}$ comprised of possibly overlapping subsets of the input space and a *benchmark hypothesis class* $\mathcal{H}$ of predictors. The objective in multi-group learning is to construct a predictor f whose accuracy is not much worse than the best (possibly different) predictor $h_g \in \mathcal{H}$ for each group $g \in \mathcal{G}$ simultaneously. This generalizes the traditional (on-average) statistical learning setting when $\mathcal{G}$ contains the entire input space. Rothblum & Yona (2021) gives an initial boosting-based algorithm that achieves multi-group learning, and Tosh & Hsu (2022) provide simpler algorithms with improved sample complexity. These results all assume no further structure on $\mathcal{G}$. In this paper, we study the natural special case in which $\mathcal{G}$ has hierarchical structure.

To this end, we present two main contributions. First, we identify a mathematically natural structural property for groups — namely, hierarchical structure — that permits a simple and efficient algorithm that achieves multi-group learning with near-optimal group-wise error rates. Second, we conduct an extensive empirical evaluation of this algorithm against other baselines for multi-group learning on several real datasets with hierarchical group structure. This partially addresses a question posed by Tosh & Hsu (2022) to design a learning algorithm that outputs a simple, deterministic predictor like their `Prepend` algorithm enjoys near-optimal error rates. The empirical results show our algorithm indeed achieves multi-group generalization

*Equal contribution   [1]Department of Computer Science, Columbia University. Correspondence to: Samuel Deng <samdeng@cs.columbia.edu>, Daniel Hsu <djhsu@cs.columbia.edu>.

on par with—and, on some groups, better than—existing multi-group learning methods. This supports our theory and suggests that, in the case where groups are hierarchically structured, the tree representation of our algorithm is a good inductive bias for generalization on the subgroups.

1.1. Summary of Results

First, we analyze two algorithms for hierarchical multi-group learning, showing "near-optimal" group-wise excess error rates for the latter. The naïve first algorithm, detailed in Section 3.1, simply applies ERM to a partition of the input space formed by the hierarchical groups. This procedure of training a separate "decoupled" predictor on disjoint subsets of the input space to achieve some desired multi-group property has been studied before (Dwork et al., 2018); we simply extend the analysis to the hierarchical multi-group learning framework.

Our main algorithm, MGL-Tree (Algorithm 1), outputs a simple decision tree predictor that is guaranteed to achieve error competitive to the best possible predictor in a benchmark hypothesis class for every group simultaneously. Importantly, the best benchmark hypothesis for each group may differ. Namely, we show that, for finite $\mathcal{H}$, MGL-Tree achieves multi-group learning with a group-wise excess error rate of $O\left(\sqrt{\log(|\mathcal{H}||\mathcal{G}|)/n_g}\right)$, where $\mathcal{H}$ is the hypothesis class, $\mathcal{G}$ is the collection of hierarchical groups, and n_g is the number of training examples for a group g. We note that this also extends to $\mathcal{H}$ of bounded VC dimension, and these guarantees hold for any bounded loss function.

In comparison, an algorithm of Tosh & Hsu (2022) also achieves the state-of-the-art group-wise excess error rate of $O\left(\sqrt{\log(|\mathcal{H}||\mathcal{G}|)/n_g}\right)$ for finite $\mathcal{H}$ and $\mathcal{G}$ (without the hierarchical restriction on $\mathcal{G}$) with a black-box online-to-batch reduction. The resulting predictor, however, is a randomized ensemble of n (also) random predictors, one for each training sample, and requires the explicit enumeration of a finite hypothesis class. It is unclear if a practical implementation of such an algorithm exists. A separate algorithm of Tosh & Hsu (2022) and Globus-Harris et al. (2022), called Prepend, on the other hand, outputs a simple, interpretable, and determnistic decision list predictor that avoids enumerating $\mathcal{H}$. Its group-wise excess error rate, however, is not optimal: $O\left(\sqrt[3]{\log(|\mathcal{H}||\mathcal{G}|)/\gamma n_g}\right)$, where γ is the probability of the group with the smallest mass. Resolving this trade-off between simplicity and optimal rates was an unresolved issue posed in Tosh & Hsu (2022). Our algorithm avoids the trade-off and achieves the state-of-the-art excess error rate with a simple predictor that refines its predictions through a breadth-first search on the tree induced by the hierarchical group structure.

Second, we conduct an empirical evaluation of existing multi-group learning algorithms on datasets with natural hierarchical group structure. We evaluate MGL-Tree, the Prepend algorithm, and the ERM baseline in several binary classification problems. We evaluate different benchmark hypothesis classes (linear predictors, decision trees, and tree ensembles) and different hierarchical collections of groups. We consider twelve US Census datasets from Ding et al. (2021) for the tasks of predicting income, health care coverage, and employment. We consider four different collections of hierarchically structured groups arising from different collections of demographic attributes: (1) race, sex, and age, (2) race, sex, and education, (3) US state, race, and sex, and (4) US state, race, and age.

On each dataset, we evaluate the generalization of each algorithm through classification accuracy on a held-out test set. We find that MGL-Tree consistently improves on the accuracy of the global ERM and group-specific ERM hypotheses from our benchmark class, validating our theory. Our algorithm also consistently achieves equal or higher accuracy than Prepend, suggesting that the decision tree representation of our final predictor is a more favorable inductive bias to hierarchically structured groups than a decision list of possibly arbitrary order. We also find that, in some cases, MGL-Tree even outperforms more complex ERM predictors (e.g., tree ensembles) which were empirically observed in previous work to have good multi-group generalization properties (Gardner et al., 2022; Pfohl et al., 2022).

1.2. Related Work

Our work touches upon several areas of theoretical and empirical literature, including multi-group learning, fairness, and hidden stratification.

Multi-group learning. The multi-group agnostic PAC learning setting was first formalized by Rothblum & Yona (2021), and initial algorithms for multi-group learning relied on a reduction to the outcome indistinguishability framework of Dwork et al. (2021). Tosh & Hsu (2022) introduced additional algorithms for achieving multi-group learning with improved, near-optimal sample complexity, discussed in Section 1.1. Both of these works, like ours, study this problem in the batch setting. Blum & Lykouris (2020) studies the online setting, where the goal is regret minimization with respect to multiple, possibly intersecting subgroups. Rittler & Chaudhuri (2024) considers the problem in the active learning setting, where the learner can decide which examples to obtain labels on. Blum et al. (2017); Haghtalab et al. (2022) study another learning model in which the learner may decide the number of samples to obtain from different distributions (which are not necessarily subsets of the input space). This setup is related to *multi-task learning*, in which samples are drawn from multiple, possibly related

tasks (i.e., distributions, which are also not necessarily subsets of the input space), and the goal is to perform well on a single test distribution while avoiding so-called "negative transfer" between tasks (Ben-David et al., 2002; Ben-David & Schuller, 2003; Mansour et al., 2008; Li et al., 2023). Multi-group agnostic PAC learning is also related to a recent strand of work in *multicalibration* for providing more stringent calibration guarantees across multiple, possibly intersecting groups (Hébert-Johnson et al., 2018; Kim et al., 2019; Globus-Harris et al., 2023).

Group and minimax fairness. A primary motivation for multi-group learning comes from the group fairness literature, where the goal is to achieve specific fairness criterion over multiple intersecting subgroups of individuals defined by characteristics such as sex or race. Hébert-Johnson et al. (2018) and Kearns et al. (2018) initiated the study of fairness across multiple *intersecting* subgroups constructed from different combinations of protected attribute values (e.g. intersections of race and gender). These works developed algorithms for various fairness notions in the algorithmic fairness literature (e.g. equalized odds or false positive parity). In the context of fairness, multi-group learning is related to the natural fairness criterion of "minimax fairness," in which the objective is to minimize the maximum loss across all groups rather than achieving parity in losses (Diana et al., 2021). We note that, when fairness is a concern, "gerrymandered," intersectional groups naturally admit a hierarchical group structure (Kearns et al., 2018) once the defining features are ordered.

Slice discovery and hidden stratification. In many other learning scenarios, fine-grained high error subgroups are unknown ahead of time. A plethora of empirical work has uncovered various settings in which predictors with high overall accuracy predict poorly on meaningful subgroups of the input space (Buolamwini & Gebru, 2018; Oakden-Rayner et al., 2020; Borkan et al., 2019). The burgeoning literature of *slice discovery* aims to develop methods that identify high-error subgroups that are not known *a priori* (Chung et al., 2019; Eyuboglu et al., 2022; Sohoni et al., 2020). These slice discovery methods find these high-error subgroups by recursively slicing features into smaller and smaller subcategories, so the collection of found subgroups are presented in a hierarchical structure. This literature serves as a motivation to our work, but we note that, in multi-group learning, the collection of groups is known ahead of time (perhaps already "discovered" as a result of one of these methods).

2. Problem Setup

2.1. Setting and notation

We are interested in the standard supervised statistical learning setting with additional group structure. We denote the input space as $\mathcal{X}$, the output space as $\mathcal{Y}$, and the decision space as $\mathcal{Z}$. It is not necessary that $\mathcal{Z} = \mathcal{Y}$. We denote $\mathcal{D}$ as an arbitrary joint probability distribution over $\mathcal{X} \times \mathcal{Y}$. We denote $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ as an i.i.d. set of random examples drawn from $\mathcal{D}$. A *predictor (hypothesis)* is a function $h : \mathcal{X} \to \mathcal{Z}$. A *benchmark hypothesis class* $\mathcal{H}$ is a collection of such functions.

We will compute probabilities and expectations with respect to the true distribution $\mathcal{D}$, or the empirical distribution over S (where a new example (x, y) is drawn uniformly at random from S). We denote these as $\mathbb{P}[\cdot]$, $\mathbb{E}[\cdot]$ for the true distribution, and $\mathbb{P}_S[\cdot]$, $\mathbb{E}_S[\cdot]$ for the empirical distribution.

For a bounded loss function $\ell : \mathcal{Z} \times \mathcal{Y} \to [0, 1]$, we denote the *risk* as $L_{\mathcal{D}}(f) := \mathbb{E}_{\mathcal{D}}[\ell(f(x), y)]$ and the *empirical risk* as $L_S(f) := \mathbb{E}_S[\ell(f(x), y)] = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i)$.

Multi-group learning notation. A *group* is a subset of the input space, $g \subseteq \mathcal{X}$. We will overload g to also denote the indicator function for the set, $g(x) := \mathbf{1}\{x \in g\}$. $\mathcal{G}$ denotes the collection of groups of interest, so it is a collection of subsets of the input space. For a set of n examples, denote n_g as the number of examples where $x \in g$, i.e. $n_g := \sum_{i=1}^{n} g(x_i)$.

For a distribution P (in our case, either $\mathcal{D}$ or S), we will write $\mathbb{P}[A \mid g]$ (respectively, $\mathbb{E}[X \mid g]$) to denote the probability of an event A (respectively, expectation of a random variable X) conditioned on the event that, on a random $X \sim P$, $g(X) = 1$ (i.e. $X \in g$).

Our main metric for comparison will be the *group-conditional risk*: $L_{\mathcal{D}}(f \mid g) := \mathbb{E}_{\mathcal{D}}[\ell(f(x), y) \mid g]$. We also denote the *group-conditional empirical risk* as $L_S(f \mid g) := \frac{1}{n_g} \sum_{i=1}^{n} g(x_i)\ell(f(x_i), y_i)$. For a group $g \in \mathcal{G}$, we use $\hat{h}^g \in \arg\min_{h \in \mathcal{H}} L_S(h \mid g)$ to refer to a hypothesis in $\mathcal{H}$ that minimizes the empirical risk on *only* the examples from g.

2.2. Background on multi-group agnostic learning

Multi-group (agnostic PAC) learning is a generalization of the traditional PAC learning setting (Valiant, 1984). The goal in multi-group learning is to construct a predictor that achieves small risk *conditional* on the membership of an example in a group, for a collection of (potentially overlapping) groups $\mathcal{G}$ simultaneously. "Small risk" is measured with respect to the best hypothesis in a *benchmark hypothesis class*, $\mathcal{H}$.

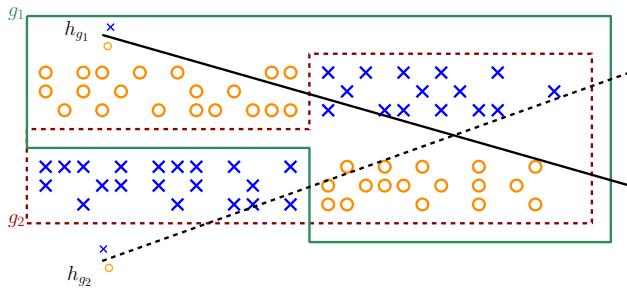A *multi-group learning algorithm* for $(\mathcal{G}, \mathcal{H})$ takes as in-

put a dataset of n i.i.d. labeled examples from a distribution $\mathcal{D}$ and outputs a predictor $f : \mathcal{X} \to \mathcal{Z}$. The goal of such an algorithm is to ensure that the *group-wise excess errors* $L_{\mathcal{D}}(f \mid g) - \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h \mid g)$ are small for all $g \in \mathcal{G}$. Formally, we say that such an algorithm achieves the *multi-group learning property* with group-wise excess error bounds of $\epsilon_g(n, \delta)$.

Definition 2.1 (Multi-group learning property). A multi-group learning algorithm for $(\mathcal{G}, \mathcal{H})$ satisfies the *multi-group learning property* with group-wise excess error bounds $\epsilon_g(\cdot, \cdot)$ for $g \in \mathcal{G}$ if it returns a predictor $f : \mathcal{X} \to \mathcal{Z}$ such that, with probability at least $1 - \delta$ over n i.i.d. examples from $\mathcal{D}$,

$$L_{\mathcal{D}}(f \mid g) - \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h \mid g) \leq \epsilon_g(n, \delta) \quad \text{for all } g \in \mathcal{G}.$$

These group-wise excess errors will all be decreasing functions of n_g, and it is straightforward to convert a group-wise excess error bound to an upper bound on the sample size n_g sufficient to guarantee a given excess error bound $\epsilon_g > 0$ for group g.

Note that f is a *single* predictor that simultaneously achieves this objective with respect to the best hypothesis in *every* group. In fact, every group may have a different best predictor achieving $\inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h \mid g)$, and there may be no single $h \in \mathcal{H}$ that achieves small risk for all groups simultaneously, as Figure 1 shows. Thus, we focus on the *improper* learning setting, where f is allowed to be outside of $\mathcal{H}$.

Figure 1. **No best h for all groups simultaneously.** Letting $\mathcal{H}$ be the class of halfspaces, the groups g_1 (indicated by the green solid line) and g_2 (indicated by the red dotted line) overlap, but their optimal predictors h_{g_1} and h_{g_2} are much different.

When $\mathcal{X} \in \mathcal{G}$, this is a generalization of the traditional agnostic PAC learning setting, where it is well-known that *empirical risk minimization* (ERM) is necessary and sufficient. Tosh & Hsu (2022) observe that, if we are *a priori* concerned with the conditional distribution on each $g \in \mathcal{G}$, then the optimal excess error from ERM on each group is $O(\sqrt{\log(|\mathcal{H}|)/n_g})$ for finite $\mathcal{H}$ and $O(\sqrt{d \log(n_g)/n_g}$ for $\mathcal{H}$ with VC dimension $d < \infty$. They show that there

exists an algorithm that achieves this at a "near-optimal" rate that incurs an extra $O\sqrt{\log |\mathcal{G}|}$ factor for finite $\mathcal{G}$, with rate $O(\sqrt{\log(|\mathcal{G}||\mathcal{H}|)/n_g})$. In Section 3.2, we give a much simpler, deterministic algorithm that achieves this rate in the case of hierarchically structured groups.

We briefly remark that, when Rothblum & Yona (2021) introduced the learning model in Definition 2.1, they required the additional assumption of "multi-PAC compatibility." This is the requirement that $\mathcal{H}$ contains a hypothesis that is nearly optimal for every group simultaneously. This precludes settings where the best hypothesis on each group can be vastly different, such as in Figure 1 or the practical settings discussed in Section 1. Therefore, we avoid this assumption and provide group-wise excess error bounds where the best $h \in \mathcal{H}$ is allowed to be different across the groups $\mathcal{G}$.

2.3. Hierarchical group structure

We note a couple important features of the collection of groups $\mathcal{G}$ in this learning model. First, $\mathcal{G}$ is fixed *a priori* and is known during both training and test. In addition, the group identity of each example is known during training and test; that is, $g(x)$ can always be evaluated. In some applications to fairness, group membership for certain "protected attributes" may not be available at training or test time due to privacy or legal concerns (Veale & Binns, 2017; Holstein et al., 2019; Lahoti et al., 2020).

Because of such restrictions, recent works in fairness have pivoted from demanding fairness on a small, fixed number of coarse demographic attributes (e.g. race, sex) to instead fixing a combinatorially large number of subgroups based on conjunctions of attributes (Kearns et al., 2018). For example, considering all the possible subgroups generated by conjunctions of race, sex, and age generates an exponentially large number of subgroups. This motivates the importance of obtaining near-optimal sample complexity rates that are *logarithmic* in $|\mathcal{G}|$, the total number of subgroups.
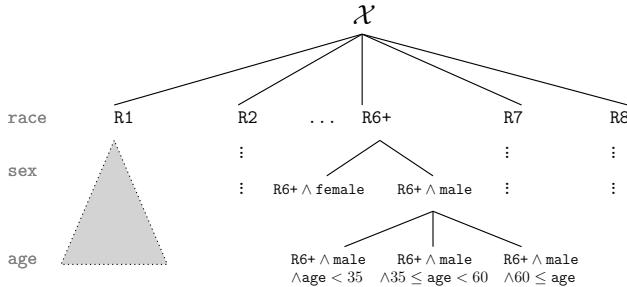
Subdividing an input space $\mathcal{X}$ based on relevant attributes in a given order generates an exponentially large number of hierarchically structured subgroups. We can also obtain a hierarchically structured collection of groups from dividing an input space top-down into categories and further subcategories. In the unsupervised machine learning context, hierarchically structured groups are obtained via, say, agglomerative clustering or a dyadic partitioning.

Definition 2.2 (Hierarchically Structured Groups). A collection of groups $\mathcal{G}$ is *hierarchically structured* if, for every pair of distinct groups $g, g' \in \mathcal{G}$, exactly one of the following holds:

1. $g \cap g' = \emptyset$ (g and g' are disjoint).

2. $g \subset g'$ (g is contained in g').

3. $g' \subset g$ (g' is contained in g).

A hierarchically structured collection of groups also naturally admits a rooted tree or collection of trees, where each group $g \in \mathcal{G}$ is a node in the tree. Figure 2 gives an example.

Figure 2. **Example of a hierarchically structured tree.** Each level of the tree above corresponds to a demographic attribute (race, sex, and age). Proceeding down the tree yields increasingly granular subgroups. The leaves are the most granular level, with subgroups such as R6+ ∧ male ∧ age < 35.

Definition 2.3 (Hierarchical Tree). Let $\mathcal{X}$ be an input space. A collection of hierarchically structured groups $\mathcal{G}$ on $\mathcal{X}$ with $\mathcal{X} \in \mathcal{G}$ (without loss of generality) admits a *(rooted) tree* $\mathcal{T}_{\mathcal{G}}$ such that each node of $\mathcal{T}_{\mathcal{G}}$ is a group g, $\mathcal{X}$ is the root of $\mathcal{T}_{\mathcal{G}}$, and, for every pair $g, g' \in \mathcal{T}_{\mathcal{G}}$:

1. If g is a child of g', then $g \subset g'$.

2. If g and g' are the same distance from the root, then $g \cap g' = \emptyset$.

Subdividing on attributes in this way can create different hierarchical trees depending on the order of attributes. For example, subdividing by race first then sex gives a different tree than subdividing on race first and then age (although the leaves are the same). In our experiments (Section 4), similar findings held for all orderings. In practice, this ordering may be decided by a domain expert.

3. Algorithms for learning hierarchical groups

In this section, we analyze two multi-group learning algorithms for hierarchically structured groups. The first, in Section 3.1, is a baseline algorithm of Dwork et al. (2018) studied in the context of fairness, but we include a brief analysis in our setting as a warm-up. It does not achieve the state-of-the-art group error rates for multi-group learning. Our main algorithm, in Section 3.2, outputs a simple decision tree predictor that obtains the near-optimal excess error rates, satisfying a desideratum from Tosh & Hsu (2022) for the case of hierarchically structured $\mathcal{G}$. We restrict attention to finite $\mathcal{H}$; infinite $\mathcal{H}$ with finite VC dimenison are considered in Appendices B and C, along with the proofs.

3.1. Algorithm: "Decoupled" Group ERM

We first analyze a simple algorithm for multi-group learning in the hierarchical setting: training a predictor on each disjoint leaf node. This procedure has been considered before in previous work in the context of fairness (Dwork et al., 2018), but we include an analysis of its sample complexity in the multi-group learning framework with hierarchical groups for completeness and comparison.

Let $\mathcal{G}$ be a hierarchically structured collection of groups, with hierarchical tree $\mathcal{T}_{\mathcal{G}}$. Let $g_1, \ldots, g_N$ be the leaves of the tree, and let $\hat{h}^{g_i} \in \arg\min_{h \in \mathcal{H}} L_S(h \mid g_i)$ be the ERM predictor trained only on samples from leaf g_i. Our predictor $f : \mathcal{X} \to \mathcal{Z}$ is simply:

$$f(x) := \hat{h}^{g_i}(x) \text{ if } x \in g_i, \tag{1}$$

which is well-defined if the leaves $g_1, \ldots, g_N$ form a partitioning of the input space $\mathcal{X}$.

We note that this "decoupled" predictor was originally proposed under the assumption that the groups partition the input space (Dwork et al., 2018), so we make that assumption here on the leaves. However, we note that our main algorithm in Section 3.2 avoids this assumption and works with general hierarchically structured groups.

Theorem 3.1. *Let $\mathcal{H}$ be a hypothesis class and let $\mathcal{G}$ be a collection of hierarchically structured groups with leaf nodes $g_1, \ldots, g_N$ partitioning the input space $\mathcal{X}$. Let $\ell(\cdot, \cdot) \in [0, 1]$ be any bounded loss function. Then, with probability $1 - \delta$ over n i.i.d. examples $(x, y) \sim \mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, f in Equation (1) satisfies the multi-group learning property:*

$$L_{\mathcal{D}}(f \mid g) - \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h \mid g) \leq \epsilon_g(n, \delta), \tag{2}$$

for any $g \in \mathcal{G}$ such that $g = \bigcup_{i=1}^{k} g_i$ and $\epsilon_g(n, \delta) := 9 \sum_{i=1}^{k} \frac{\mathbb{P}[x \in g_i]}{\sum_{j=1}^{k} \mathbb{P}[x \in g_j]} \sqrt{\frac{\log(8|\mathcal{G}||\mathcal{H}|/\delta)}{n_{g_i}}}$.

This simple algorithm achieves multi-group learning, but it is not optimal. In the worst case (e.g., when all disjoint leaf nodes have the same probability mass), the error rate grows with $\sqrt{|\mathcal{G}|}$, which is far larger than the $\sqrt{\log|\mathcal{G}|}$ achievable by other methods, as discussed in Section 2.3.

3.2. Algorithm: Multi-group Tree

In this section, we present MGL-Tree (Algorithm 1), which achieves the group-wise excess error of $O(\sqrt{\log|\mathcal{H}||\mathcal{G}|/n_g})$ with a simple and interpretable decision tree predictor. One can view MGL-Tree as an adaptation of the Prepend algorithm of Tosh & Hsu (2022) and Globus-Harris et al. (2022). Our analysis of MGL-Tree depends on the hierarchical structure of $\mathcal{G}$ and is very different from that of

`Prepend`, leading to the improved near-optimal sample complexity rate.

One unresolved issue stated in Tosh & Hsu (2022) was to find an algorithm for multi-group learning that is both "simple" and enjoys near-optimal sample complexity. If we were only concerned with the conditional distribution of a single fixed group, standard statistical learning theory suggests that, for finite $\mathcal{H}$, ERM is necessary and sufficient to achieve an excess error of $\epsilon_g = O(\sqrt{\log(|\mathcal{H}|)/n_g})$. Tosh & Hsu (2022) gives an algorithm (different from `Prepend`) that achieves the near-optimal excess error of $\epsilon_g = O(\sqrt{\log(|\mathcal{H}||\mathcal{G}|)/n_g})$. The resulting predictor, however, is a randomized ensemble of n (also) random predictors, one for each training sample, and requires the enumeration of a finite hypothesis class. On the other hand, `Prepend` outputs a simple, interpretable, and determnistic decision list predictor that avoids enumerating $\mathcal{H}$. Its excess error, however, is not optimal: $\epsilon_g = O(\sqrt[3]{\log(|\mathcal{H}||\mathcal{G}|)/(\gamma n_g)})$, where $\gamma := \inf_{g \in \mathcal{G}} \mathbb{P}_{\mathcal{D}}[x \in g]$. This trade-off between simplicity and optimal rates was an unresolved issue stated in Tosh & Hsu (2022). For hierarchical $\mathcal{G}$, `MGL-Tree` avoids this trade-off between simplicity (and implementability) and near-optimal rates, as stated in Theorems 3.2 and C.3. This allows us to also conduct an empirical evaluation of `MGL-Tree` in Section 4.

Theorem 3.2. *Suppose $\mathcal{H}$ is a finite benchmark hypothesis class and $\mathcal{G}$ is a collection of hierarchically structured groups. Let $\ell(\cdot, \cdot) \in [0, 1]$ be any bounded loss function. Then, with probability $1 - \delta$ over n i.i.d. examples $(x, y) \sim \mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, `MGL-Tree` taking input*

$$\epsilon_n(g) := 18\sqrt{\frac{2\log(|\mathcal{G}||\mathcal{H}|) + \log(8/\delta)}{n_g}}$$

outputs a predictor f satisfying the multi-group learning property with:

$$L_{\mathcal{D}}(f \mid g) - \min_{h \in \mathcal{H}} L(h \mid g) \leq 2\epsilon_n(g) \quad \textit{for all } g \in \mathcal{G}. \quad (3)$$

If $\mathcal{H}$ is infinite, with finite VC dimension $d \geq 1$, then setting `MGL-Tree` with

$$\epsilon_n(g) := 18\sqrt{\frac{2d\log(16|\mathcal{G}|n/\delta)}{n_g}}$$

outputs a predictor f satisfying Equation (3).

`MGL-Tree` gradually refines the "decision tree" predictor f by updating its behavior on children groups when a child's predictor is significantly better than its parent's by margin $\epsilon_n(g)$. Each node g in $\mathcal{T}_{\mathcal{G}}$ has an associated predictor f^g. To evaluate f on the input x, we find the "deepest" $g \in \mathcal{T}_{\mathcal{G}}$ that contains x by following a path starting from the root $\mathcal{X}$ and

---

Algorithm 1 `MGL-Tree`

Require:
1: S, a training dataset.
2: Collection of hierarchically structured groups $\mathcal{G} \subseteq 2^{\mathcal{X}}$.
3: Error rates $\epsilon_n(g) \in (0, 1)$ for all $g \in \mathcal{G}$

Ensure: Decision tree $f : \mathcal{X} \to \mathcal{Z}$.
4: Order $\mathcal{G}$ into a *hierarchical tree* $\mathcal{T}_{\mathcal{G}}$ (Definition 2.3).
5: Initialize the root: $f^{\mathcal{X}} := \hat{h}^{\mathcal{X}}$.
6: **for** each node $g \in \mathcal{T}_{\mathcal{G}} \setminus \{\mathcal{X}\}$ in breadth-first order **do**
7:     Compute: $\text{err}_g := \mathbb{E}_S[\ell(f^g(x), y) \mid x \in g] - \mathbb{E}_S[\ell(\hat{h}^g(x), y) \mid x \in g] - \epsilon_n(g)$.
8:     **if** $\text{err}_g \geq 0$ **then**
9:         Set $f^g := \hat{h}^g$.
10:     **else**
11:         Set $f^g := f^{\text{pa}(g)}$, where $\text{pa}(g)$ denotes the parent node of g.
12:     **end if**
13: **end for**
14: **return** $f : \mathcal{X} \to \mathcal{Z}$, a decision tree predictor.

---

moving from parent to child whenever the child contains x. The prediction $f(x)$ is taken to be $f^g(x)$.

The main tension that the algorithm resolves is whether to use a predictor for a coarser-grained group (say, the predictor for just `R6+`) vs. a predictor for a finer-grained group (say, the predictor for `R6+` $\wedge$ `male` $\wedge$ `age` < 35). The finer-grained predictor may be more specific to the finer-grained group, but the finer-grained group necessarily had less samples to train on, and, hence, higher variance. The coarser-grained predictor may be less specific to the finer-grained group (and might, therefore, not pick up on the group-specific signal for the labels), but it had more samples to train on. By the hierarchical structure, coarser predictors are always the parents of finer-grained children.

The main intuition of the algorithm is to resolve this trade-off by only using the finer-grained group's predictor when it is significantly better than its coarser-grained parent, where the "significance" is $\epsilon_n(g)$ in the Step 7 of the algorithm. Lemma 3.3 below establishes a nice "monotonicity" property — every update operation will never make the overall decision tree violate any error bounds it satisfied earlier in the search, so we always make forward progress towards a decision tree that satisfies the multi-group learning objective.

To prove the correctness of `MGL-Tree`, we need to show the main property that the tree's predictions monotonically improve as `MGL-Tree` runs. This monotonicity property is similar in spirit to the key idea in the analysis of `Prepend` (Tosh & Hsu, 2022), but we need to exploit breadth-first search ordering on the hierarchical tree to attain this monotonicity property. We present the statements of the key

lemma for proving correctness, and the explicit statement of correctness here in the main body. The full proof is in Appendix C.

Lemma 3.3. *Consider any step of Algorithm 1 where we are considering $g \in \mathcal{G}$. Let f_{old} be the decision tree at this step before updating (the state of the tree at line 7). Let $\hat{h}^g \in \arg\min_{h \in \mathcal{H}} L_S(h \mid g)$ for all $g \in \mathcal{G}$. Then, for all $x \in g$, $f_{\text{old}}(x) = h^{g'}(x)$ for some $g' \supset g$ already visited by the algorithm.*

Theorem 3.4 (Correctness of `MGL-Tree`). *Let $\mathcal{G}$ be a hierarchically structured collection of groups, let $S = \{(x_i, y_i)\}_{i=1}^n$ be n i.i.d. training data drawn from any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, and let $\epsilon_n : \mathcal{G} \to (0, 1)$ be any error rate function. Then, Algorithm 1 run on these parameters outputs a predictor $f : \mathcal{X} \to \mathcal{A}$ satisfying:*

$$L_S(f \mid g) \leq \inf_{h \in \mathcal{H}} L_S(h \mid g) + \epsilon_n(g), \quad \text{for all } g \in \mathcal{G}.$$

This predictor also has the bonus that it is easily interpretable. By reading off the nodes of the decision tree predictor, one can interpret the source of model errors for a specific group from its subgroups deeper in the tree. For example, for a hierarchical collection of groups induced by `race`, `sex`, and `age`, one can check if errors on `race` groups stem from an intersection of `race ∧ sex` or the deeper intersection of `race ∧ sex ∧ age` one level down the tree. Because the guarantees from multi-group learning apply to *any* arbitrary bounded loss function, this property may be attractive in, say, fairness scenarios that demand some notion of fairness that can be encoded by a bounded loss function. We leave such applications for future work.

4. Empirical evaluation

In this section, we perform empirical evaluations of Algorithm 1 for several classification problems with natural hierarchical group structure.

4.1. Experiment setup and learning task

Models. For the benchmark hypothesis class $\mathcal{H}$, we consider the following models: logistic regression, decision trees (with maximum depth 2, 4, and 8), random forests, and XGBoost. Implementation details are in Appendix E.

Dataset overview. We conduct our experiments on twelve U.S. Census datasets from the Folktables package of Ding et al. (2021). These datasets include many demographic attributes or other protected features of individuals, and they are large enough to have intersections with substantial data even when subdividing on multiple attributes.

Nine datasets come from each of the U.S. states of New York (NY), California (CA), and Florida (FL), where we consider the Employment, Income, and Coverage binary

classification tasks. The other three nationwide datasets come from concatenating the data from 18 U.S. states, determined by the two most populous states in each U.S. Census designated region of the U.S. (U.S. Census, 2023) and considering the same three binary classification tasks. The binary classification tasks are as follows:

- **Income.** Predict whether employed individuals older than 16 make over $50,000 USD per year.
- **Employment.** Predict whether individuals older than 16 and younger than 90 are employed.
- **Coverage.** Predict whether individuals younger than 65 (ineligible for Medicare) with income less than $30,000 USD has coverage from public health insurance.

For each of the nine state datasets, we consider two hierarchically structured collections of intersecting groups that arise naturally from subdividing demographic attributes of $\{\texttt{race}, \texttt{sex}, \texttt{age}\}$ or $\{\texttt{race}, \texttt{sex}, \texttt{edu}\}$. For the three nationwide datasets, we divide on $\{\texttt{state}, \texttt{race}, \texttt{age}\}$ or $\{\texttt{state}, \texttt{race}, \texttt{sex}\}$. In fairness settings, by defining the intersecting groups of interest in this way, we interpolate between a coarse, small collection of pre-defined groups from a single attribute to ensuring fairness for individuals (Kearns et al., 2018). Additional details are provided in Appendix F.

Setup. For each benchmark hypothesis class, we consider four training procedures:

- **ERM.** The predictor from minimizing empirical risk over all the data: $f \in \arg\min_{f \in \mathcal{H}} L_S(f)$.
- **`Prepend`.** The decision list predictor output by `Prepend`. Details are given in Appendix D.
- **`MGL-Tree`.** The decision tree from Algorithm 1.
- **Group ERM.** The group-specific ERM predictor $\hat{h}^g \in \arg\min_{f \in \mathcal{H}} L_S(f \mid g)$ (when evaluating performance on group g).

For all ERM procedures, we optimize the logistic loss, a typical surrogate for the misclassification loss. We note that "Group ERM" is a benchmark procedure for if we were *a priori* concerned only about the conditional distribution for group $g \in \mathcal{G}$ (as discussed in Section 2.2).

We evaluate the multi-group generalization of each predictor by measuring misclassification error on data restricted to each group from a held-out test set of 20% of the data. We repeat over 10 trials and plot the mean and standard errors.

4.2. Main findings

The main findings of our empirical evaluations are:

1. **`MGL-Tree` achieves multi-group learning.**
2. **`MGL-Tree` performs on par with or better than `Prepend` on all groups across all datasets.**
3. **`MGL-Tree` outperforms "subgroup robust" baseline**

methods on certain groups.

4. **Simpler hypothesis classes lead to larger improvements when updating the predictor of `MGL-Tree`.**

We note that these findings are consistent across all twelve datasets. Due to space, we relegate most of the plots to Appendix H and present the results on a couple of the datasets (CA Employment and CA Income) in the main body.

First, we empirically observe that `MGL-Tree` indeed achieves multi-group learning. This supports the upper bound on excess error of Theorem 3.2. Across all datasets, `MGL-Tree` performs on-par with or better than the minimum error achieved by the ERM or Group ERM predictors from the benchmark hypothesis class. For instance, for CA Employment, Figure 3 shows that `MGL-Tree` (labeled "TREE") has consistently lower error than both ERM and Group ERM for logistic regression, depth 2 decision trees, and random forests. In cases where `MGL-Tree` beats ERM by a significant amount, the predictor from `MGL-Tree` has identified a high-error subgroup on a specific slice of the population, which may prove useful for further auditing.

Though `MGL-Tree` consistently matches or beats ERM and Group ERM on other datasets, the gaps in group generalization are sometimes not as stark; this may suggest that ERM and Group ERM with the benchmark hypothesis class already achieve close to the Bayes optimal risk. Some evidence supports this: on such datasets, even random forest or XGBoost tree ensembles – which are known to be very effective on tabular datasets (Gardner et al., 2022) – have comparable error to simpler benchmark classes.

Second, `MGL-Tree` performs on par with or better than `Prepend` across almost all groups across all datasets. This suggests that, in the case of hierarchially structured groups, `MGL-Tree` is better at finding an ordering of group predictors that improves the accuracy of the model. This might not be surprising, as `MGL-Tree` directly exploits an inductive bias towards hierarchically structured groups by constructing its decision tree with knowledge of the hierarchial structure. `Prepend`, on the other hand, constructs a decision list with a possibly arbitrary ordering of groups and group predictors. We observe this in Figure 3 for the CA Income dataset, with further discussion in Section 4.3.

Third, `MGL-Tree` sometimes outperforms more complex random forest and XGBoost baselines even when using relatively simple benchmark hypothesis classes. A couple of recent works suggest that simply running ERM with a tree ensemble method such as random forest or XGBoost achieves good multi-group performance (Pfohl et al., 2022; Gardner et al., 2022). For example, we observed that `MGL-Tree` run with the simple benchmark hypothesis class of linear predictors (specifically, logistic regression models) yields a substantial improvement over the ERM random forest base-

line on almost all groups in the CA Employment dataset. See Figure 5 in Appendix H for more details.
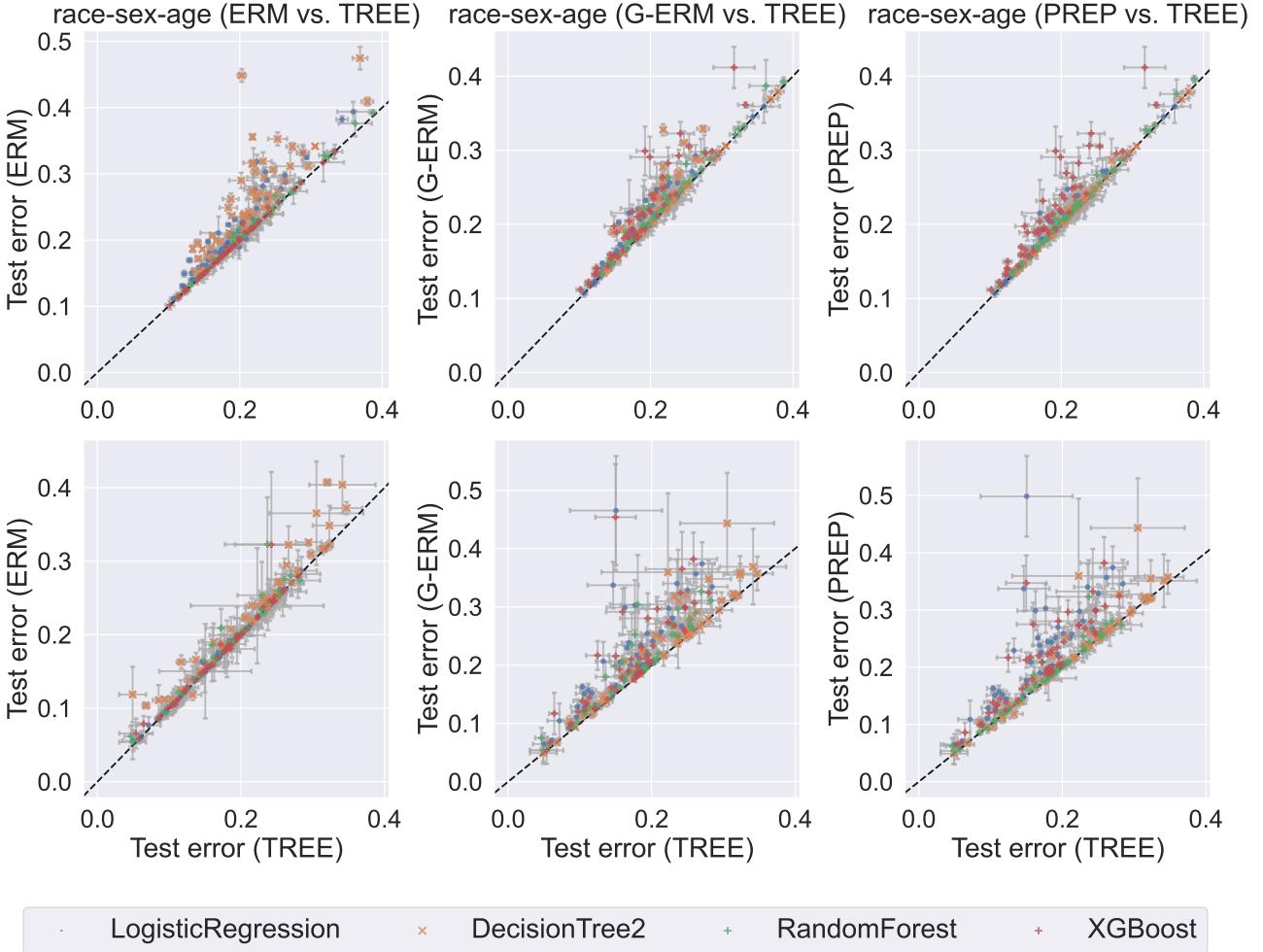
Finally, we also observe that, for simpler model classes, the predictor from `MGL-Tree` yields a larger improvement over baseline ERM and Group ERM training compared to more complex model classes. We see this in both datasets in Figure 3, where the simplest model class of depth-2 decision trees has the largest gaps from the $y = x$ line. The more complex hypothesis class of linear models from logistic regression yields points closer to the line. The most complex classes of random forests and XGBoost see even less improvement than both logistic regression and depth-2 decision trees from running `MGL-Tree`, as evident from how many points hug the $y = x$ line. This suggests that for the predictor from `MGL-Tree`, improvements in subgroup generalization compared to the benchmark hypothesis class becomes more difficult as the class becomes more expressive. This behavior is also predicted from the dependence on $\mathcal{H}$ in our excess error rates in Theorem 3.2.

4.3. Comparing `Prepend` and `MGL-Tree`

One main finding in Section 4.2 was that `MGL-Tree` generalizes on par with or better than `Prepend` on all groups across all datasets. Figure 3 shows two examples of this, and Appendix H contains more examples. In this section, we compare the final predictors output by `Prepend` and `MGL-Tree` on two datasets from our empirical study when the benchmark class is logistic regression. We find that, perhaps unsurprisingly, `MGL-Tree` achieves better generalization than `Prepend` because it prefers using the group-specific hypotheses for coarser groups at higher levels in the tree, while `Prepend` tends to overfit to the finer-grained subgroups at the leaves. We summarize these findings here and refer the reader to Appendix G for the full breakdown of the final predictors from `MGL-Tree` and `Prepend`.

CA Employment. From the upper-right plot in Figure 3, we observe that, for most groups, `MGL-Tree` performs on par with `Prepend` when both are instantiated with logistic regression (blue dots). There are still several groups where `MGL-Tree` performs better, as illustrated by the blue dots above the $y = x$ line, but the vast majority of the groups are on par with `Prepend`.

Upon a closer examination of the group-specific predictors used by `Prepend` or `MGL-Tree`, we find that, for CA Employment, `Prepend` and `MGL-Tree` mostly resort to using leaf node predictors. In the case of the `race-sex-age` groups, the leaves correspond to conjunctions of three attributes such as `R6+` $\wedge$ `male` $\wedge$ `age` < 35. As such, the final predictor for both algorithms is almost identical, but both show a marked improvement over simply using ERM for most groups, as shown in the upper-left plot in Figure 3.

Figure 3. **Test accuracy on `race-sex-age` groups for CA Employment (top row) and CA Income (bottom row).** Each point in the plot represents the test error on a specific group. The $y = x$ line represents equal error between our `MGL-Tree` and the competing method; points *above* the $y = x$ line are groups where `MGL-Tree` exhibits better generalization.

CA Income. From the lower-right plot in Figure 3, on the other hand, we see that `MGL-Tree` generalizes better than `Prepend` on many groups. Upon inspection of the final predictor, we observe that `MGL-Tree` employs coarser-grained subgroup predictors further up the tree (e.g. using `ALL` or `R1`) on many of the leaves, while, on most groups, `Prepend` predicts with finer-grained subgroup predictors.

This suggests that the arbitrary order of the `Prepend` decision list can overfit to finer-grained subgroups that may not have sufficient samples for the finer-grained predictors to generalize. On the other hand, `MGL-Tree` only predicts with a finer-grained subgroup's predictor when its coarser-grained parent has sufficiently high error due to its breadth-first search on the hierarchical tree.

5. Conclusion

In this paper, we study the multi-group (agnostic PAC) learning framework in the natural case where the groups are hierarchically structured. We give an algorithm that achieves multi-group learning by constructing a decision tree of predictors for each group in the hierarchical collection of groups. This algorithm achieves multi-group learning with a near-optimal group-wise excess error of $O(\sqrt{\log(|\mathcal{H}||\mathcal{G}|)/n_g})$ and outputs a simple and deterministic predictor, addressing an issue posed by (Tosh & Hsu, 2022) for the case of hierarchically structured groups. We then conduct an extensive empirical evaluation of our algorithm and find that it achieves attractive generalization properties on real datasets with hierarchical group structure, as the theory suggests.

Acknowledgements

Impact Statement

This work focuses on a learning setting initially inspired by the fairness literature (see, e.g., Diana et al. (2021), Hébert-Johnson et al. (2018), Kearns et al. (2018), Buolamwini & Gebru (2018)) requiring some fairness criterion to be met on multiple, possibly overlapping subgroups of individuals. If our algorithms are used in the fairness setting, it is important to note that the jury is still out on what definition of algorithmic fairness is philosophically and legally sound. Our particular learning objective corresponds to ensuring that every subgroup is as well-off as the best possible hypothesis for that subgroup; however, this is at odds with a fairness definition such as, for instance, statistical parity (Barocas et al., 2023). We also note that our setting may be used in applications more general than fairness, when fine-grained accuracy is the only concern. In those cases, the usual societal implications of the application area apply.

References

Balsubramani, A., Dasgupta, S., Freund, y., and Moran, S. An adaptive nearest neighbor rule for classification. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Barocas, S., Hardt, M., and Narayanan, A. *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, 2023.

Ben-David, S. and Schuller, R. Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines: 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003. Proceedings*, pp. 567–580. Springer, 2003.

Ben-David, S., Gehrke, J., and Schuller, R. A theoretical framework for learning from a pool of disparate data sources. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 443–449, 2002.

Bissoto, A., Fornaciali, M., Valle, E., and Avila, S. (de)constructing bias on skin lesion datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

Blum, A. and Lykouris, T. Advancing Subgroup Fairness via Sleeping Experts. In Vidick, T. (ed.), *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 55:1–55:24, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-134-4. doi: 10.4230/LIPIcs.ITCS.2020.55. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2020.55.

Blum, A., Haghtalab, N., Procaccia, A. D., and Qiao, M. Collaborative PAC Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Borkan, D., Dixon, L., Sorensen, J., Thain, N., and Vasserman, L. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pp. 491–500, 2019.

Buolamwini, J. and Gebru, T. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, pp. 77–91. PMLR, January 2018. URL https://proceedings.mlr.press/v81/buolamwini18a.html. ISSN: 2640-3498.

Chen, T. and Guestrin, C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, August 2016. doi: 10.1145/2939672.2939785. URL http://arxiv.org/abs/1603.02754. arXiv:1603.02754 [cs].

Chung, Y., Kraska, T., Polyzotis, N., Tae, K. H., and Whang, S. E. Slice finder: Automated data slicing for model validation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 1550–1553. IEEE, 2019.

De Vries, T., Misra, I., Wang, C., and Van der Maaten, L. Does object recognition work for everyone? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 52–59, 2019.

DeGrave, A. J., Janizek, J. D., and Lee, S.-I. AI for radiographic COVID-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, July 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00338-7. URL https://www.nature.com/articles/s42256-021-00338-7. Number: 7 Publisher: Nature Publishing Group.

Diana, E., Gill, W., Kearns, M., Kenthapadi, K., and Roth, A. Minimax group fairness: Algorithms and experiments. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 66–76, 2021.

Ding, F., Hardt, M., Miller, J., and Schmidt, L. Retiring adult: New datasets for fair machine learning. *Advances in neural information processing systems*, 34:6478–6490, 2021.

Dwork, C., Immorlica, N., Kalai, A. T., and Leiserson, M. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on fairness, accountability and transparency*, pp. 119–133. PMLR, 2018.

Dwork, C., Kim, M. P., Reingold, O., Rothblum, G. N., and Yona, G. Outcome indistinguishability. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 1095–1108, 2021.

Eyuboglu, S., Varma, M., Saab, K., Delbrouck, J.-B., Lee-Messer, C., Dunnmon, J., Zou, J., and Ré, C. Domino: Discovering systematic errors with cross-modal embeddings. *arXiv preprint arXiv:2203.14960*, 2022.

Gardner, J., Popovic, Z., and Schmidt, L. Subgroup robustness grows on trees: An empirical baseline investigation. *Advances in Neural Information Processing Systems*, 35: 9939–9954, 2022.

Globus-Harris, I., Kearns, M., and Roth, A. An algorithmic framework for bias bounties. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1106–1124, 2022.

Globus-Harris, I., Harrison, D., Kearns, M., Roth, A., and Sorrell, J. Multicalibration as boosting for regression. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Haghtalab, N., Jordan, M., and Zhao, E. On-Demand Sampling: Learning Optimally from Multiple Distributions. *Advances in Neural Information Processing Systems*, 35: 406–419, December 2022.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016.

Hébert-Johnson, U., Kim, M., Reingold, O., and Rothblum, G. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pp. 1939–1948. PMLR, 2018.

Holstein, K., Vaughan, J. W., Daumé III, H., Dudík, M., and Wallach, H. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–16, May 2019. doi: 10.1145/3290605.3300830. URL http://arxiv.org/abs/1812.05239. arXiv:1812.05239 [cs].

Kearns, M., Neel, S., Roth, A., and Wu, Z. S. Preventing Fairness Gerrymandering: Auditing and Learning for Subgroup Fairness. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2564–2572. PMLR, July 2018. ISSN: 2640-3498.

Kim, M. P., Ghorbani, A., and Zou, J. Multiaccuracy: Black-Box Post-Processing for Fairness in Classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '19, pp. 247–254, New York, NY, USA, January 2019. Association for Computing Machinery. ISBN 978-1-4503-6324-2. doi: 10.1145/3306618.3314287. URL https://dl.acm.org/doi/10.1145/3306618.3314287.

Kuehlkamp, A., Becker, B., and Bowyer, K. Gender-from-iris or gender-from-mascara? In *2017 IEEE winter conference on applications of computer vision (WACV)*, pp. 1151–1159. IEEE, 2017.

Lahoti, P., Beutel, A., Chen, J., Lee, K., Prost, F., Thain, N., Wang, X., and Chi, E. Fairness without Demographics through Adversarially Reweighted Learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 728–740. Curran Associates, Inc., 2020.

Li, D., Nguyen, H., and Zhang, H. R. Identification of negative transfers in multitask learning using surrogate models. *Transactions on Machine Learning Research*, 2023.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation with multiple sources. In Koller, D., Schuurmans, D., Bengio, Y., and Bottou, L. (eds.), *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL https://proceedings.neurips.cc/paper_files/paper/2008/file/0e65972dce68dad4d52d063967f0a705-Paper.pdf.

Oakden-Rayner, L., Dunnmon, J., Carneiro, G., and Ré, C. Hidden stratification causes clinically meaningful failures in machine learning for medical imaging. In *Proceedings of the ACM conference on health, inference, and learning*, pp. 151–159, 2020.

Orr, L., Leszczynski, M., Arora, S., Wu, S., Guha, N., Ling, X., and Re, C. Bootleg: Chasing the tail with self-supervised named entity disambiguation. *arXiv preprint arXiv:2010.10363*, 2020.

Pfohl, S. R., Zhang, H., Xu, Y., Foryciarz, A., Ghassemi, M., and Shah, N. H. A comparison of approaches to improve worst-case predictive model performance over patient subpopulations. *Scientific Reports*, 12(1): 3254, February 2022. ISSN 2045-2322. doi: 10.1038/ s41598-022-07167-7. URL https://www.nature. com/articles/s41598-022-07167-7. Number: 1 Publisher: Nature Publishing Group.

Rittler, N. and Chaudhuri, K. Agnostic multi-group active learning. *Advances in Neural Information Processing Systems*, 36, 2024.

Rothblum, G. N. and Yona, G. Multi-group agnostic pac learnability. In *International Conference on Machine Learning*, pp. 9107–9115. PMLR, 2021.

Sohoni, N., Dunnmon, J., Angus, G., Gu, A., and Ré, C. No subclass left behind: Fine-grained robustness in coarse-grained classification problems. *Advances in Neural Information Processing Systems*, 33:19339–19352, 2020.

Tosh, C. J. and Hsu, D. Simple and near-optimal algorithms for hidden stratification and multi-group learning. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 21633–21657. PMLR, June 2022. ISSN: 2640-3498.

U.S. Census. U.S. Census Bureau Regions and Divisions of the United States, December 2023. URL https: //www2.census.gov/geo/pdfs/maps-data/ maps/reference/us_regdiv.pdf.

Valiant, L. G. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984. ISSN 0001-0782. doi: 10.1145/1968.1972. URL https:// dl.acm.org/doi/10.1145/1968.1972.

Varma, M., Orr, L., Wu, S., Leszczynski, M., Ling, X., and Ré, C. Cross-Domain Data Integration for Named Entity Disambiguation in Biomedical Text. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t. (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4566–4575, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp. 388. URL https://aclanthology.org/2021. findings-emnlp.388.

Veale, M. and Binns, R. Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data. *Big Data & Society*, 4(2):2053951717743530, December 2017. ISSN 2053-9517. doi: 10.1177/ 2053951717743530. URL https://doi.org/10. 1177/2053951717743530. Publisher: SAGE Publications Ltd.

A. Technical Lemmas

The main technical lemma we will employ in our proofs is a result from Tosh & Hsu (2022) that establishes the uniform convergence of risks conditional on group membership. This generalizes a lemma from Balsubramani et al. (2019) that gives a rate for the uniform convergence of empirical conditional measures.

Lemma A.1 (Theorem 1 from Tosh & Hsu (2022)). *Let $\mathcal{H}$ be a hypothesis class, let $\mathcal{G}$ be a collection of groups, and let $\ell : \mathcal{Z} \times \mathcal{Y} \to [0, 1]$ be a bounded loss function. Given an i.i.d. sample of size n drawn from a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, with probability at least $1 - \delta$,*

$$|L_{\mathcal{D}}(h \mid g) - L_S(h \mid g)| \leq 9\sqrt{\frac{2 \log(S_{2n}(\overline{\mathcal{H}})S_{2n}(\mathcal{G})) + \log(8/\delta)}{n_g}},$$

for all $h \in \mathcal{H}$ and $g \in \mathcal{G}$, where $S_k(C)$ is the kth shattering coefficient of the collection C of sets, and $\overline{\mathcal{H}} := \{(x, y) \mapsto \mathbf{1}\{\ell(h(x), y) \geq t\} : h \in \mathcal{H}, t \in \mathbb{R}\}$ is the set of (boolean) functions constructed from composing the hypothesis class $\mathcal{H}$ with the loss $\ell(\cdot, \cdot)$.

Two immediate consequences of Lemma A.1 are Lemmas A.2 and A.3.

Lemma A.2. *In the setting of Lemma A.1, if $\mathcal{H}$ and $\mathcal{G}$ are finite, then*

$$|L_{\mathcal{D}}(h \mid g) - L_S(h \mid g)| \leq 9\sqrt{\frac{2 \log(|\mathcal{H}||\mathcal{G}|) + \log(8/\delta)}{n_g}}$$

for all $h \in \mathcal{H}$ and all $g \in \mathcal{G}$.

Proof. For a finite set, the shattering coefficient is bounded by the size of the set, so $S_{2n}(\overline{\mathcal{H}}) \leq |\mathcal{H}|$ and $S_{2n}(\mathcal{G}) \leq |\mathcal{G}|$. $\square$

Lemma A.3. *In the setting of Lemma A.1, if $\mathcal{H}$ has VC dimension $d > 0$, then*

$$|L_{\mathcal{D}}(h \mid g) - L_S(h \mid g)| \leq 9\sqrt{\frac{2d \log(2|\mathcal{G}|n) + \log(8/\delta)}{n_g}}.$$

Proof. By Sauer's lemma, $S_{2n}(\overline{\mathcal{H}}) \leq \sum_{i=0}^{d} \binom{n}{i} \leq (2n)^d$. $\square$

We will also need the following two lemmas for decomposing group conditional risks for hierarchically structured $\mathcal{G}$.

Lemma A.4. *Let $g_1, \ldots g_N$ be N disjoint groups, and let $S = \{(x_i, y_i)\}_{i=1}^n$ be a dataset of n i.i.d. examples. Let $n_{\cup g}$ be the number of examples in $\bigcup_{k=1}^N g_k$, and let n_{g_k} be the number of examples in g_k. Then, the group conditional empirical risk for $\bigcup_{k=1}^K g_k$ decomposes as follows:*

$$L_S\left(f \,\middle|\, \bigcup_{k=1}^N g_k\right) = \sum_{k=1}^N \frac{n_{g_k}}{n_{\cup g}} L_S(f \mid g_k). \tag{4}$$

Proof. By definition of group conditional empirical risk,

$$L_S\left(f \,\middle|\, \bigcup_{k=1}^N g_k\right) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\left\{x_i \in \bigcup_{k=1}^N g_k\right\} \ell(f(x_i), y_i).$$

We note that $g_1, \ldots, g_N$ are disjoint, so:

$$\begin{aligned}
\frac{1}{n} \sum_{i=1}^n \mathbf{1}\left\{x_i \in \bigcup_{k=1}^N g_k\right\} \ell(f(x_i), y_i) &= \frac{1}{n_{\cup g}} \sum_{i=1}^n \left(\sum_{k=1}^N g_k(x_i)\ell(f(x_i), y_i)\right) \\
&= \frac{1}{n_{\cup g}} \sum_{k=1}^N \left(\sum_{i=1}^n g_k(x_i)\ell(f(x_i), y_i)\right) \\
&= \frac{1}{n_{\cup g}} \sum_{k=1}^N n_{g_k} L_S(f \mid g_k) = \sum_{k=1}^N \frac{n_{g_k}}{n_{\cup g}} L_S(f \mid g_k).
\end{aligned}$$

13

In the first equality, the $g_k(x)$ are just indicators for disjoint groups, so that immediately follows from boolean algebra on $\mathbf{1}\left\{x \in \bigcup_{k=1}^{N} g_k\right\}$. The second equality just switches order of summation, and the third is the definition of group conditional empirical risk again. □

Lemma A.5. *Let $g_1, \ldots g_N$ be N disjoint groups. Then, the group conditional risk for $\bigcup_{k=1}^{N} g_k$ decomposes:*

$$L_{\mathcal{D}}\left(f \mid \bigcup_{k=1}^{N} g_k\right) = \sum_{k=1}^{N} \frac{\mathbb{P}[x \in g_k]}{\sum_{j=1}^{K} \mathbb{P}[x \in g_j]} L_{\mathcal{D}}\left(f \mid g_k\right). \tag{5}$$

Proof. By definition of group conditional risk,

$$L_{\mathcal{D}}(f \mid g) = \mathbb{E}\left[\ell(f(x), y) \mid x \in g\right].$$

We first claim that $L_{\mathcal{D}}(f \mid g) = \frac{1}{\mathbb{P}[x \in g]} \mathbb{E}\left[g(x)\ell(f(x), y)\right]$. This follows from:

$$
\begin{aligned}
\mathbb{E}\left[g(x)\ell(f(x), y)\right] &= \mathbb{E}\left[\mathbb{E}\left[g(x)\ell(f(x), y) \mid x \in g\right]\right] \\
&= \mathbb{E}\left[g(x)\mathbb{E}\left[\ell(f(x), y) \mid x \in g\right]\right] \\
&= \mathbb{P}[x \in g]\mathbb{E}\left[\ell(f(x), y) \mid x \in g\right] \\
&= \mathbb{P}[x \in g]L_{\mathcal{D}}(f \mid g).
\end{aligned}
$$

Using this fact, we can re-write the group conditional risk as:

$$L_{\mathcal{D}}\left(f \mid \bigcup_{k=1}^{N} g_k\right) = \frac{1}{\mathbb{P}\left[x \in \bigcup_{j=1}^{N} g_j\right]} \mathbb{E}\left[\mathbf{1}\left\{x \in \bigcup_{k=1}^{N} g_k\right\} \ell(f(x), y)\right].$$

Because $g_1, \ldots, g_N$ are disjoint, we can use additivity of $\mathbb{P}(\cdot)$:

$$
\begin{aligned}
&= \frac{1}{\sum_{j=1}^{N} \mathbb{P}\left[x \in g_j\right]} \mathbb{E}\left[\sum_{k=1}^{N} g_k(x)\ell(f(x), y)\right] \\
&= \frac{1}{\sum_{j=1}^{N} \mathbb{P}[x \in g_j]} \sum_{k=1}^{N} \mathbb{E}\left[g_k(x)\ell(f(x), y)\right]
\end{aligned}
$$

Using the same fact that $\mathbb{P}\left[x \in g_k\right] L_{\mathcal{D}}\left(f \mid g_k\right) = \mathbb{E}\left[g_k(x)\ell(f(x), y)\right]$, we get the desired result:

$$= \sum_{k=1}^{N} \frac{\mathbb{P}\left[x \in g_k\right]}{\sum_{j=1}^{N} \mathbb{P}\left[x \in g_j\right]} L_{\mathcal{D}}(f \mid g_k).$$

□

B. Proof of Theorem 3.1

We give the full version of Theorem 3.1 here, including the case where $\mathcal{H}$ has finite VC dimension.

Theorem B.1. *Let $\mathcal{H}$ be a hypothesis class with VC dimension $d > 0$ and let $\mathcal{G}$ be a collection of hierarchically structured groups with leaf nodes $g_1, \ldots, g_N$ partitioning the input space $\mathcal{X}$. Let $\ell(\cdot, \cdot) \in [0, 1]$ be any bounded loss function. Then, with probability $1 - \delta$ over n i.i.d. examples $(x, y) \sim \mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, f in Equation (1) satisfies the multi-group learning property with*

$$L_{\mathcal{D}}\left(f \mid g\right) - \inf_{h \in \mathcal{H}} L_{\mathcal{D}}\left(h \mid g\right) \le 9 \sum_{i=1}^{k} \frac{\mathbb{P}[x \in g_i]}{\sum_{j=1}^{k} \mathbb{P}[x \in g_j]} \sqrt{\frac{2d \log(16|\mathcal{G}|n/\delta)}{n_{g_i}}} \tag{6}$$

for any $g \in \mathcal{G}$ such that $g = \bigcup_{i=1}^{k} g_i$. For finite $\mathcal{H}$,

$$L_{\mathcal{D}}\left(f \mid g\right) - \inf_{h \in \mathcal{H}} L_{\mathcal{D}}\left(h \mid g\right) \le 9 \sum_{i=1}^{k} \frac{\mathbb{P}[x \in g_i]}{\sum_{j=1}^{k} \mathbb{P}[x \in g_j]} \sqrt{\frac{\log(8|\mathcal{G}||\mathcal{H}|/\delta)}{n_{g_i}}}. \tag{7}$$

Proof. We first show the proof for $\mathcal{H}$ with finite VC dimension d.

For each disjoint atomic group g_i, for $i \in [N]$, the behavior of f is simply to use $\hat{h}_{g_i}$. We know from uniform convergence of conditional risks (Lemma A.3) that

$$L_{\mathcal{D}}(f \mid g_i) = L_{\mathcal{D}}(\hat{h}_{g_i} \mid g_i) \leq \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h \mid g_i) + 9\sqrt{\frac{2d \log(16|\mathcal{G}|n/\delta)}{n_{g_i}}}, \tag{8}$$

where n_{g_i} is the number of examples, all from group g_i. Consider the union of k of the N disjoint atomic groups, $\bigcup_{i=1}^{k} g_i$ (without loss of generality for index i, let g_i be any one of the N groups). By Lemma A.5,

$$L_{\mathcal{D}}\left(f \,\middle|\, \bigcup_{i=1}^{k} g_i\right) = \sum_{i=1}^{k} \frac{\mathbb{P}(g_i)}{\sum_{j=1}^{k} \mathbb{P}(g_j)} L_{\mathcal{D}}(f \mid g_i).$$

By Equation (8) and the definition of the predictor f from Section 3.1,

$$L_{\mathcal{D}}\left(f \,\middle|\, \bigcup_{i=1}^{k} g_i\right) = \sum_{i=1}^{k} \frac{\mathbb{P}(g_i)}{\sum_{j=1}^{k} \mathbb{P}(g_j)} L_{\mathcal{D}}(f \mid g_i) = \sum_{i=1}^{k} \frac{\mathbb{P}(g_i)}{\sum_{j=1}^{k} \mathbb{P}(g_j)} L_{\mathcal{D}}(\hat{h}_{g_i} \mid g_i)$$

$$\leq \sum_{i=1}^{k} \frac{\mathbb{P}(g_i)}{\sum_{j=1}^{k} \mathbb{P}(g_j)} \left(L_{\mathcal{D}}(h_{g_i}^* \mid g_i) + 9\sqrt{\frac{2d \log(16|\mathcal{G}|n/\delta)}{n_{g_i}}}\right),$$

where $h_{g_i}^* \in \arg\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h \mid g_i)$. Now, let $h_{\cup g_i}^* \in \arg\min_{h^* \in \mathcal{H}} L_{\mathcal{D}}(h^* \mid \bigcup_{i=1}^{k} g_i)$. Because each $h_{g_i}^*$ is optimal for their conditional risks on their respective group g_i,

$$\leq \sum_{i=1}^{k} \frac{\mathbb{P}(g_i)}{\sum_{j=1}^{k} \mathbb{P}(g_j)} \left(L_{\mathcal{D}}(h_{\cup g_i}^* \mid g_i) + 9\sqrt{\frac{2d \log(16|\mathcal{G}|n/\delta)}{n_{g_i}}}\right)$$

$$= \sum_{i=1}^{k} \frac{\mathbb{P}(g_i)}{\sum_{j=1}^{k} \mathbb{P}(g_j)} L_{\mathcal{D}}(h_{\cup g_i}^* \mid g_i) + \sum_{i=1}^{k} \frac{\mathbb{P}(g_i)}{\sum_{j=1}^{k} \mathbb{P}(g_j)} 9\sqrt{\frac{2d \log(16|\mathcal{G}|n/\delta)}{n_{g_i}}}$$

$$= L_{\mathcal{D}}\left(h_{\cup g_i}^* \,\middle|\, \bigcup_{i=1}^{k} g_i\right) + \sum_{i=1}^{k} \frac{\mathbb{P}(g_i)}{\sum_{j=1}^{k} \mathbb{P}(g_j)} 9\sqrt{\frac{2d \log(16|\mathcal{G}|n/\delta)}{n_{g_i}}},$$

whcih is our our result. The final equality is from applying Lemma A.5 again to combine the first term.

The proof for finite $\mathcal{H}$ is identical, but use the finite $\mathcal{H}$ uniform convergence bound for conditional risks in Lemma A.2 instead of Equation (8). $\square$

C. Proof of Theorems 3.2 and C.3

We now present the proof of MGL-Tree (Algorithm 1). This algorithm outputs a final decision tree predictor, $f : \mathcal{X} \to \mathcal{Z}$. Each node of the decision tree is a group $g \in \mathcal{G}$ with an associated *working predictor* $f^g : \mathcal{X} \to \mathcal{Z}$. The goal of the algorithm is to determine a good working predictor for each node of the tree. For each group $g \in \mathcal{G}$ and an i.i.d. sample S, we'll denote $\hat{h}^g$ to be the ERM minimizer of group conditional empirical risk:

$$\hat{h}^g := \arg\min_{h \in \mathcal{H}} L_S(h \mid g)$$

We construct the decision tree as follows. First, we generate the hierarchical tree $\mathcal{T}_{\mathcal{G}}$ (Definition 2.3) from the hierarchically structured collection of groups $\mathcal{G}$. For simplicity, the root is $\mathcal{X}$. In this tree, every node g is a subset of its ancestors. We begin by initializing the root node's working predictor $f^{\mathcal{X}} := \hat{h}^{\mathcal{X}}$, the ERM minimizer of group conditional empirical risk for all of $\mathcal{X}$ — this is just standard unconditional empirical risk $L_S(h)$.

To assign working predictors f^g to each node, we start from the root of the tree where $g = \mathcal{X}$ and visit all $|\mathcal{G}|$ nodes in the tree in breadth-first order. Let $f^{\mathrm{pa}(g)}$ denote the working predictor for the parent of node g. The main idea is to set the

working predictor at node g to $f^g := \hat{h}^g$ only if its parent is insufficient for achieving the desired margin of error $\epsilon_n(g)$. Otherwise, node g inherits its working predictor from its parent: $f^g := f^{\mathrm{pa}(g)}$. To show that Algorithm 1 is correct, the key is to prove a "monotonicity" property: at each update operation, the algorithm does not violate any error bounds for groups further up the tree.

Let f_{old} denote the state of the decision tree *before* an update operation. In Algorithm 1, this corresponds to the state of the decision tree at line 7 in each iteration of the main BFS loop. We will analyze f_{old} in the proofs of Theorem C.2 and our main Theorems 3.2 and C.3.

We state one more obvious lemma concerning the behavior of f when we choose to inherit the parent's working predictor, $f^{\mathrm{pa}(g)}$. When we do this, f is functionally equivalent to f_{old} on group g and all the nodes on the path from g back up to the root. This is referred to as 3.3 in the main body, and is restated here as Lemma C.1 for convenience.

Lemma C.1 (Behavior of f_{old}). *Consider any step of Algorithm 1 where we are considering $g \in \mathcal{G}$. Let f_{old} be the decision tree at this step before updating (the state of the tree at line 7). Let $\hat{h}^g \in \arg\min_{h \in \mathcal{H}} L_S(h \mid g)$ for all $g \in \mathcal{G}$. Then, for all $x \in g$, $f_{\mathrm{old}}(x) = h^{g'}(x)$ for some $g' \supset g$ already visited by the algorithm.*

Proof. This just follows by induction. For the first step of Algorithm 1, f_{old} is simply $h \in \arg\min_{h \in \mathcal{H}} L_S(h)$, the ERM predictor over all of $\mathcal{X}$. Of course, $g \subset \mathcal{X}$ for any g. Assume the lemma for all $g' \in \mathcal{G}'$, the set of already visited nodes. Suppose we are on step $g \in \mathcal{G}$ in our BFS. Then, if $x \in g$, by hierarchical structure and BFS, $x \in g'$ for some $g \subset g'$ because we've visited all parents before their children. Therefore, f_{old} uses $h^{g'}$ for some $g' \supset g$. $\qquad\square$

Lemma C.1 allows us to apply Lemma A.1, our conditional uniform convergence bound, on f_{old}, as it is functionally equivalent to some $h \in \mathcal{H}$, our benchmark hypothesis class.

The key to Algorithm 1 is that each update operation at any node g does not make f violate the error bounds it satisfied further up the tree. We observe that, at an update iteration (when $\mathrm{err}_g \geq 0$), either we accept the (conditional) ERM predictor $f^g := \hat{h}^g$ or we inherit the parent's working predictor $f^g := f^{\mathrm{pa}(g)}$. See Figure 4.
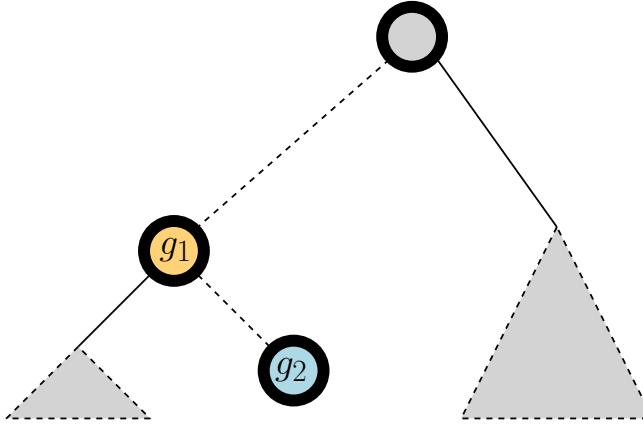

Figure 4. Let g_1, the yellow node, be a group that Algorithm 1 has already seen. Suppose f updates on g_2. We see that g_1 is on the path from g_2 to the root. We need to show that the inequality for g_1 is not violated after the update.

The next theorem, Theorem C.2, establishes the correctness of the algorithm when it terminates. This is stated in the main body as Theorem 3.4, and we state it here as Theorem C.2.

Theorem C.2 (Correctness of `MGL-Tree`). *Let $\mathcal{G}$ be a hierarchically structured collection of groups, let $S = \{(x_i, y_i)\}_{i=1}^n$ be n i.i.d. training data drawn from any distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, and let $\epsilon_n : \mathcal{G} \to (0, 1)$ be any error rate function. Then, Algorithm 1 run on these parameters outputs a predictor $f : \mathcal{X} \to \mathcal{A}$ satisfying:*

$$L_S(f \mid g) \leq \inf_{h \in \mathcal{H}} L_S(h \mid g) + \epsilon_n(g), \quad \text{for all } g \in \mathcal{G}. \tag{9}$$

16

Proof. Consider any $g^* \in \mathcal{G}$, which corresponds to a node in the decision tree, f. We analyze the step of the algorithm's breadth-first search concerned with g^* and argue that this step does not violate any inequalities of the form (9) satisfied up until this step. This is sufficient to prove correctness because g^* is an arbitrary node, and the tree traversal will visit every node, so (9) will be satisfied for all nodes in the tree.

On the current step for g^*, the algorithm can either decide to update or not. If the if-condition is not satisfied and it does not update, then we keep $f := f_{\text{old}}$ from the previous round, and we are done. All inequalities previously satisfied must continue to be satisfied because f did not change.

Suppose, then, that the algorithm *did* update. Let $P_{g^*} := \{\hat{g}_1, \ldots, \hat{g}_k\}$ be the set of nodes on the path from g^* to the root of the tree, including the root. Then, for all nodes $g \notin P_{g^*}$, f continues to satisfy (9) because $g \cap g^* = \emptyset$, so for $x \in g$, $f(x) = f_{\text{old}}(x)$, as before. This is due to the hierarchical structure — any node *not* on the path from g^* back up to the root must be disjoint from g^*.

Now, consider our final case: any $\hat{g}_j \in P_{g^*}$ for $j \in [k]$, a node back up to the root from g^*. Again, we are in the case where we updated, so f has changed. By the hierarchical structure, if $\hat{g}_j$ is further up the tree from g^*, then $g^* \subset \hat{g}_j$. We need to show that $L_S(f \mid \hat{g}_j) \leq L_S(f_{\text{old}} \mid \hat{g}_j)$. Denote $\overline{g}$ as the complement of g, and apply Lemma A.4:

$$
\begin{aligned}
L_S(f \mid \hat{g}_j) &= L_S(f \mid (\hat{g}_j \cap g^*) \cup (\hat{g}_j \cap \overline{g^*})) \\
&= L_S(f \mid \hat{g}_j \cap g^*) \frac{\Pr[x \in \hat{g}_j \cap g^*]}{\Pr[x \in \hat{g}_j]} + L_S(f \mid \hat{g}_j \cap \overline{g^*}) \frac{\Pr[x \in (\hat{g}_j \cap \overline{g^*})]}{\Pr[x \in \hat{g}_j]} \\
&= L_S(f \mid g^*) \Pr[x \in g^* \mid x \in \hat{g}_j] + L_S(f \mid \hat{g}_j \cap \overline{g^*}) \Pr[x \in \overline{g^*} \mid x \in \hat{g}_j] \\
&= L_S \Pr[x \in g^* \mid x \in \hat{g}_j] + L_S(f_{\text{old}} | \hat{g}_j \cap \overline{g^*}) \Pr[x \in \overline{g^*} \mid x \in \hat{g}_j].
\end{aligned}
\tag{10}
$$

The last equality is a result of how the f operates as a decision tree. Observe that, on $x \in g^*$, our updated decision tree f uses $h^* \in \arg\min_{h \in \mathcal{H}} L_S(h \mid g^*)$. On $x \in \hat{g}_j \cap \overline{g^*}$, our decision tree f simply operates as it did before the update: $f(x) = f_{\text{old}}(x)$. By definition of h^* as ERM predictor,

$$
L_S(h^* \mid g^*) \leq L_S(f_{\text{old}} \mid g^*),
\tag{11}
$$

so combining (10) and (11),

$$
\begin{aligned}
L_S(f \mid \hat{g}_j) &= L_S(h^* \mid g^*) \Pr[x \in g^* \mid x \in \hat{g}_j] + L_S(f_{\text{old}} \mid \hat{g}_j \cap \overline{g^*}) \Pr[x \in \overline{g^*} \mid x \in \hat{g}_j] \\
&\leq L_S(f_{\text{old}} \mid g^*) \Pr[x \in g^* \mid x \in \hat{g}_j] + L_S(f_{\text{old}} \mid \hat{g}_j \cap \overline{g^*}) \Pr[x \in \overline{g^*} \mid x \in \hat{g}_j] \\
&= L_S(f_{\text{old}} \mid \hat{g}_j) \leq \min_{h \in \mathcal{H}} L_S(h \mid \hat{g}_j) + \epsilon_n(\hat{g}_j),
\end{aligned}
\tag{12}
$$

where the final equality in (12) follows from another application of Lemma A.4. Because

$$
L_S(f \mid \hat{g}_j) \leq \min_{h \in \mathcal{H}} L_S(h \mid \hat{g}_j) + \epsilon_n(\hat{g}_j),
\tag{13}
$$

where f is the updated decision list, we see that f does not violate any of the inequalities for the nodes $\hat{g}_j$ in the path up to the root, finishing our proof. $\square$

We state the result for $\mathcal{H}$ with finite VC dimension to accompany Theorem 3.2 here.

Theorem C.3. *For $\mathcal{H}$ with VC dimension bounded by $d > 0$ and the other conditions of Theorem 3.2, running Algorithm 1 with input*

$$
\epsilon_n(g) := 18 \sqrt{\frac{2d \log(16|\mathcal{G}|n/\delta)}{n_g}}
$$

outputs a predictor f that achieves

$$
L_{\mathcal{D}}(f \mid g) \leq \inf_{h \in \mathcal{H}} L(h \mid g) + 36 \sqrt{\frac{2d \log(16|\mathcal{G}|n/\delta)}{n_g}} \quad \text{for all } g \in \mathcal{G}.
\tag{14}
$$

We may now prove Theorem 3.2 and Theorem C.3 for the multi-group learning guarantee of Algorithm 1, `MGL-Tree`.

Proof. We will show this by induction on each iteration (visited group g) of Algorithm 1 for finite $\mathcal{G}$ and $\mathcal{H}$. Condition on the event that we drew our i.i.d. dataset of size n and Lemma A.2 holds. For ease of notation, we will denote

$$UC(g) := 9\sqrt{\frac{2\log(|\mathcal{G}||\mathcal{H}|) + \log(8/\delta)}{n_g}},$$

so, for all groups $g \in \mathcal{G}$, we have, uniformly over $h \in \mathcal{H}$:

$$|L_{\mathcal{D}}(h \mid g) - L_S(h \mid g)| \leq UC(g).$$

Note that we choose $\epsilon_n(g) = 2UC(g)$. Our goal will thus be to show that, for all $g \in \mathcal{G}$, the decision tree f satisfies

$$L_{\mathcal{D}}(f \mid g) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h \mid g) + 4UC(g). \tag{15}$$

For the base case, we just need to show that our starting decision tree f, where each node is initialized with $h_0 \in \arg\min_{h \in \mathcal{H}} L_S(h)$, satisfies inequality (15) for $g = \mathcal{X}$. Note that this is just standard unconditional risk. This is immediately true from standard uniform convergence and f using the ERM predictor h_0, so

$$L_{\mathcal{D}}(f) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + 4UC(g).$$

For the inductive hypothesis, assume that we are on the step of the BFS in Algorithm 1 concerned with a group $g \in \mathcal{G}$. Denote $\mathcal{G}'$ as the nodes that we already visited in our BFS, and f_{old} as the decision list *before* the possible update, as before. Then,

$$L_{\mathcal{D}}(f_{\text{old}} \mid g') \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h \mid g') + 4UC(g') \tag{16}$$

holds for all $g' \in \mathcal{G}'$.

To prove the induction, we aim to show that (15) is true for our current iteration's node g as well, regardless of whether we updated f or not. Let f be the decision list *after* the update step of Algorithm 1, and let $\hat{h}^g \in \arg\min_{h \in \mathcal{H}} L_S(h \mid g)$. We want to show, for all $h \in \mathcal{H}$:

$$L_{\mathcal{D}}(f \mid g) - L_{\mathcal{D}}(h \mid g) \leq 4UC(g) \tag{17}$$
$$L_{\mathcal{D}}(f \mid g') - L_{\mathcal{D}}(h \mid g') \leq 4UC(g'), \text{ for all } g' \in \mathcal{G}'. \tag{18}$$

So, showing (17) and (18) are our goals for each of our two cases: whether we update or not. These two cases depend on whether $L_S(f_{\text{old}} \mid g) - L_S(\hat{h}^g \mid g) \leq \epsilon_n(g)$ or $L_S(f_{\text{old}} \mid g) - L_S(\hat{h}^g \mid g) > \epsilon_n(g)$, the central comparison in our algorithm.

Suppose we are in the first case, when we *do not* update. Because $\text{err}_g \leq 0$, we have $L_S(f_{\text{old}} \mid g) - L_S(\hat{h}^g \mid g) \leq \epsilon_n(g)$. Then, because we do not update, $f = f_{\text{old}}$, so (18) is immediately fulfilled because f is functionally equivalent to f_{old}, which, by induction satisfied the inequality already for all $g' \in \mathcal{G}'$. It suffices to show (17) for this case. Fix any $h \in \mathcal{H}$. First, with Lemma C.1, we can apply conditional uniform convergence on $f = f_{\text{old}}$. Then,

$$
\begin{aligned}
L_{\mathcal{D}}(f \mid g) - L_{\mathcal{D}}(h \mid g) &= L_{\mathcal{D}}(f_{\text{old}} \mid g) - L_{\mathcal{D}}(h \mid g) \\
&\leq L_S(f_{\text{old}} \mid g) - L_{\mathcal{D}}(h \mid g) + UC(g) \\
&\leq L_S(f_{\text{old}} \mid g) - L_S(h \mid g) + 2UC(g) \\
&\leq L_S(f_{\text{old}} \mid g) - L_S(\hat{h}^g \mid g) + 2UC(g) \\
&\leq \epsilon_n(g) + 2UC(g) = 4UC(g).
\end{aligned}
$$

The first inequality is from Lemma C.1 and Lemma A.2. The third inequality is from the fact that $\hat{h}^g$ is the optimal ERM predictor conditioned on $x \in g$. This proves (17) for the first case where we do not update f.

Suppose we are in the second case. In this case, we *do* update and we have that $L_S(f_{\text{old}} \mid g) - L_S(\hat{h}^g \mid g) > \epsilon_n(g)$. In this case, f_{old} is the decision tree before the update, and f is the tree after the update. Its *working predictor* has been updated to $\hat{h}^g$. Immediately, we have $L_S(f \mid g) - L_S(\hat{h}^g \mid g) = 0$ for the current node g, so, for any $h \in \mathcal{H}$,

$$
\begin{aligned}
L_{\mathcal{D}}(f \mid g) - L_{\mathcal{D}}(h \mid g) &= L_{\mathcal{D}}(\hat{h}^g \mid g) - L_{\mathcal{D}}(h \mid g) \\
&\leq L_S(\hat{h}^g \mid g) - L_S(h \mid g) + 2UC(g) \\
&\leq 2UC(g) \leq 4UC(g).
\end{aligned}
$$

The first equality is because f is functionally equivalent to $\hat{h}^g$ on all $x \in g$, and the first inequality comes from applying Lemma A.2 twice to get sample risk for h and $\hat{h}^g$. This proves (17). It suffices to prove (18). Consider any $g' \in \mathcal{G}'$, the set of already visited groups. There are two types nodes in G': g'_p, the nodes on the path back up to the root from g, and g'_{np}, the nodes *not* on the path back up to the root from g.

For any g'_{np}, by hierarchical structure, $g'_{np} \cap g = \emptyset$. So, for all $x \in g'_{np}$, the predictor just outputs as it did before the update: $f = f_{\text{old}}$. Then, for any $h \in \mathcal{H}$, we maintain the same guarantee we had before, fulfilling (18) for all g'_{np}:

$$L_{\mathcal{D}}(f \mid g'_{np}) - L_{\mathcal{D}}(h \mid g'_{np}) = L_{\mathcal{D}}(f_{\text{old}} \mid g'_{np}) - L_{\mathcal{D}}(h \mid g'_{np}) \le 4UC(g'_{np}).$$

To finish the proof, it suffices to show that (18) is still fulfilled for all g'_p, the nodes on a path back up to the root from g. Again, fix some $h \in \mathcal{H}$. Using Lemma A.5:

$$
\begin{aligned}
L_{\mathcal{D}}(f \mid g'_p) &= L_{\mathcal{D}}(f \mid (g'_p \cap g) \cup (g'_p \setminus g)) \\
&= \frac{\Pr[x \in (g'_p \cap g)]}{\Pr[x \in g'_p]} L_{\mathcal{D}}(f \mid g'_p \cap g) + \frac{\Pr[x \in (g'_p \setminus g)]}{\Pr[x \in g'_p]} L_{\mathcal{D}}(f \mid g'_p \setminus g) \\
&= \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} L_{\mathcal{D}}(f \mid g) + \frac{\Pr[x \in (g'_p \setminus g)]}{\Pr[x \in g'_p]} L_{\mathcal{D}}(f \mid g'_p \setminus g).
\end{aligned}
$$

The last equality comes from $g \subseteq g'_p$ because all nodes are contained in their ancestors. The updated f now uses $\hat{h}^g$ for all $x \in g$. Therefore:

$$L_{\mathcal{D}}(f \mid g'_p) = \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} L_{\mathcal{D}}(\hat{h}^g \mid g) + \frac{\Pr[x \in (g'_p \setminus g)]}{\Pr[x \in g'_p]} L_{\mathcal{D}}(f_{\text{old}} \mid g'_p \setminus g).$$

Apply Lemma A.2 for conditional uniform convergence on $\hat{h}^g$:

$$L_{\mathcal{D}}(f \mid g'_p) \le \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} L_S(\hat{h}^g \mid g) + \frac{\Pr[x \in (g'_p \setminus g)]}{\Pr[x \in g'_p]} L_{\mathcal{D}}(f_{\text{old}} \mid g'_p \setminus g) + \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} UC(g).$$

Adding and subtracting $\epsilon_n(g)$ to use the fact that we are in the update case where $L_S(f_{\text{old}} \mid g) > L_S(\hat{h}^g \mid g) + \epsilon_n(g)$:

$$
\begin{aligned}
&= \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} L_S(\hat{h}^g \mid g) + \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} \epsilon_n(g) \\
&\quad + \frac{\Pr[x \in (g'_p \setminus g)]}{\Pr[x \in g'_p]} L_{\mathcal{D}}(f_{\text{old}} \mid g'_p \setminus g) + \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} UC(g) - \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} \epsilon_n(g) \\
&\le \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} L_S(f_{\text{old}} \mid g) + \frac{\Pr[x \in (g'_p \setminus g)]}{\Pr[x \in g'_p]} L_{\mathcal{D}}(f_{\text{old}} \mid g'_p \setminus g) + \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} UC(g) - \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} \epsilon_n(g).
\end{aligned}
$$

Finally, using Lemma C.1 on f_{old} on $x \in g$, applying Lemma A.2 again, and recombining terms with Lemma A.5,

$$
\begin{aligned}
&\le \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} L_{\mathcal{D}}(f_{\text{old}} \mid g) + \frac{\Pr[x \in (g'_p \setminus g)]}{\Pr[x \in g'_p]} L_{\mathcal{D}}(f_{\text{old}} \mid g'_p \setminus g) + \frac{2\Pr[x \in g]}{\Pr[x \in g'_p]} UC(g) - \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} \epsilon_n(g) \\
&= L_{\mathcal{D}}(f_{\text{old}} \mid g'_p) + \frac{2\Pr[x \in g]}{\Pr[x \in g'_p]} UC(g) - \frac{\Pr[x \in g]}{\Pr[x \in g'_p]} \epsilon_n(g) \le L_{\mathcal{D}}(h \mid g'_p) + 4UC(g'_p).
\end{aligned}
$$

The final line follows by our choice of $\epsilon_n(g) = 2UC(g)$ and the inductive hypothesis on g'_p. This shows (18) for the second case where we update f, and thus completes our proof.

The proof for $\mathcal{H}$ with VC dimension $d > 0$ is identical, but with

$$UC(g) = 9\sqrt{\frac{2d \log(16|\mathcal{G}|n/\delta)}{n_g}}$$

and $\epsilon_n(g) = 2UC(g)$. $\qquad\square$

D. Description of `Prepend` (Algorithm 1 of Tosh & Hsu (2022))

In this section, we give a brief description of the `Prepend` Algorithm of Tosh & Hsu (2022), which is closely related to the concurrent decision list algorithm of Globus-Harris et al. (2022). This algorithm outputs a decision list of predictors $h \in \mathcal{H}$, where the decision nodes are groups $g \in \mathcal{G}$. Quoting from Tosh & Hsu (2022), such a decision list of length T predicts as follows on input $x \in \mathcal{X}$ by following:

$$\textbf{if } g_T(x) = 1 \textbf{ then return } h_T(x) \textbf{ else if } g_{T-1}(x) = 1 \textbf{ then return } h_{T-1}(x) \textbf{ else if } \ldots \textbf{ else return } h_0(x).$$

We shall represent such a decision list by alternating groups and hypotheses; a decision list of length T can be represented as: $f_T := [g_T, h_T, g_{T-1}, h_{T-1}, \ldots, g_1, h_1, h_0]$. To construct such a predictor, the `Prepend` algorithm maintains a decision list f_t and proceeds in rounds $t = 1, 2, 3, \ldots$ At round t, the algorithm checks whether there exists a group-hypothesis pair (g, h) that violates a desired empirical error bound. If such a pair exists, the algorithm "prepends" the group and hypothesis to the front of the list; if no such pair exists, the algorithm terminates.

Algorithm 2 displays the algorithm as presented in Tosh & Hsu (2022).

Algorithm 2 `Prepend`

Require:
 1: S, a training dataset.
 2: Collection of groups $\mathcal{G} \subseteq 2^{\mathcal{X}}$.
 3: Error rates $\epsilon_n(g) \in (0, 1)$ for all $g \in \mathcal{G}$
Ensure: Decision list $f_T : \mathcal{X} \to \mathcal{Z}$.
 4: Compute $h_0 \in \arg\min_{h \in \mathcal{H}} L_S(h)$
 5: **for** t = 0, 1, 2, ..., **do**
 6:     Compute:
$$(g_{t+1}, h_{t+1}) \in \arg\max_{(g,h) \in \mathcal{G} \times \mathcal{H}} L_S(f_t \mid g) - L_S(h \mid g) - \epsilon_n(g).$$

 7:     **if** $L_S(f_t \mid g_{t+1}) - L_S(h_{t+1} \mid g) \geq \epsilon_n(g_{t+1})$ **then**
 8:         Prepend (g_{t+1}, h_{t+1}) to f_t to obtain

$$f_{t+1} := [g_{t+1}, h_{t+1}, g_t, h_t, \ldots, g_1, h_1, h_0].$$

 9:     **else**
10:         **return** $f_t : \mathcal{X} \to \mathcal{Z}$, a decision list predictor.
11:     **end if**
12: **end for**

E. Experiment hyperparameters

In this section, we include the hyperparameters used for training each model class. Logistic regression, decision trees, and random forests all used the implementation from `scikit-learn` [1]. XGBoost was implemented using the open-source `xgboost` implementation of Chen & Guestrin (2016) [2]. All the hyperparameters not listed were set to the default values in `scikit-learn` or `xgboost`.

Model	Hyperparameters
Logistic Regression	`loss = log_loss`, `dual=False`, `solver=lbfgs`
Decision Tree	`criterion = log_loss`, `max_depth = {2, 4, 8}`
Random Forest	`criterion = log_loss`
XGBoost	`objective = binary:logistic`

Table 1. Hyperparamter settings for each choice of benchmark hypothesis class.

---

[1] https://scikit-learn.org/stable/
[2] https://xgboost.readthedocs.io/en/stable/

F. Additional dataset details

Our experiments span twelve datasets from the Folktables package of Ding et al. (2021). These are comprised of nine *statewide* datasets and three *nationwide* datasets.

Statewide datasets. We consider nine different datasets corresponding to one of the three U.S. states of New York (NY), California (CA), and Florida (FL) and one of the three binary prediction tasks of Employment, Income, and Coverage. The description of each task is found in Section 4.1.

We consider a couple of hierarchical group structures from subdividing the following categorical demographic attributes:

- `race`. 6 total categories: `R1` (White alone), `R2` (Black or African American alone), `R3+` (Native), `R6+` (Asian or Pacific Islander), `R7` (other race alone), and `R8` (two or more races).

- `sex`. 2 total categories: `M` (Male) and `F` (Female).

- `age`. 3 total categories. For Income and Employment: `Ya` (age < 35), `Ma` ($35 \leq$ age < 60), and `Oa` (age ≥ 60). For Coverage, `Ma` is instead defined as $35 \leq$ age < 50 and `Oa` is instead defined as age ≥ 50 because Coverage only concerns individuals whose age is less than 65.

- `edu`. 4 total categories: `HS-` (no high school diploma), `HS` (high school diploma or equivalent, but no college degree), `COL` (associates or bachelor's degree), and `COL+` (master's, professional, or doctorate degree beyond bachelor's).

For each of the nine statewide datasets, we subdivide on each of the demographic attributes to attain hierarchically structured groups. We focus on two collections of groups:

- **Attributes:** $\{$`race`, `sex`, `age`$\}$. Total of 54 subgroups (6 groups from `race`, 12 groups from `race` $\wedge$ `sex`, and 36 groups from `race` $\wedge$ `sex` $\wedge$ `age`).

- **Attributes:** $\{$`race`, `sex`, `edu`$\}$. Total of 66 subgroups (6 groups from `race`, 12 groups from `race` $\wedge$ `sex`, and 48 groups from `race` $\wedge$ `sex` $\wedge$ `edu`).

Tables 2 and 3 lists all the nine statewide datasets and the number of examples in each separate demographic attribute.

Dataset	All	R1	R2	R3+	R6+	R7	R8
CA Emp.	376035	231232	18831	3531	58147	46316	17978
NY Emp.	196104	137179	25362	735	16304	11089	5435
FL Emp.	196828	155682	26223	589	5356	4328	4650
CA Inc.	190187	118212	8656	1585	31039	23285	7410
NY Inc.	101270	72655	11970	340	8639	5407	2259
FL Inc.	94507	75218	11978	267	2892	2314	1838
CA Cov.	145994	82486	8652	1626	21820	24526	6884
NY Cov.	71379	44632	11262	332	7244	5754	2155
FL Cov.	73406	53841	12872	274	2350	2235	1834

Table 2. Number of examples for the `race` attribute, for each of the nine datasets. "Emp." stands for Employment, "Inc." stands for Income, and "Cov." stands for Coverage.

Nationwide datasets. We consider three more datasets constructed from the Folktables package with many more samples than the statewide datasets. For each of the classification tasks (Income, Employment, and Coverage), we gather all examples from a total of 18 U.S. states. 2 states were chosen for each of the 9 federally designated Census geographic divisions of the United States (U.S. Census, 2023).

These states and their corresponding geographic regions are: MA, CT (New England), NY, PA (Mid-Atlantic), IL, OH (East North Central), MO, MN (West North Central), FL, GA, (South Atlantic), TN, AL (East South Central), TX, LA (West South Central), AZ, CO (Mountain), CA, and WA (Pacific).

For each of the three nationwide datasets, we subdivide on each of the demographic attributes to attain hierarchically structured groups. We focus on two collections of groups:

Dataset	M	F	Ya	Ma	Oa	HS−	HS	COL	COL+
CA Emp.	185603	190432	164324	124293	87418	126469	132156	81714	35696
NY Emp.	94471	101633	81007	64422	50675	57694	71694	43783	22933
FL Emp.	95281	101547	70632	63329	62867	53825	80479	44754	17770
CA Inc.	101125	89062	65087	97507	27593	23840	80594	58824	26929
NY Inc.	51408	49862	33349	51543	16378	8417	42215	33038	17600
FL Inc.	48623	45884	28007	49451	17049	8150	44394	30289	11674
CA Cov.	65287	80707	75484	33174	37336	43646	70617	25871	5860
NY Cov.	31218	40161	36683	15235	19461	17649	35723	14282	3725
FL Cov.	32686	40720	33552	16604	23250	17209	38354	14913	2930

Table 3. Number of examples for the `sex`, `age`, and `edu` attributes, for each of the nine datasets. "Emp." stands for Employment, "Inc." stands for Income, and "Cov." stands for Coverage.

- **Attributes:** {`state`, `race`, `sex`}. Total of 342 subgroups (18 groups from `state`, 108 groups from `state` ∧ `race`, and 216 groups from `state` ∧ `race` ∧ `sex`).

- **Attributes:** {`state`, `race`, `age`}. Total of 450 subgroups (18 groups from `state`, 108 groups from `state` ∧ `race`, and 324 groups from `state` ∧ `race` ∧ `age`).

G. Comparison of `Prepend` and `MGL−Tree` (Algorithm 1) on CA Employment and CA Income

In Table 4 and Table 5, we include the group-specific predictors of Algorithm 1 (labeled "TREE") and `Prepend` (labeled "PREP") for each leaf node in the hierarchical group structure induced by splitting by `race`, `sex`, and `age` for the CA Employment and CA Income datasets. Because the leaf nodes partition the input space, this accounts for the behavior of the predictors on any possible test example $x \in \mathcal{X}$. As described in Section 4.3, we observe that Algorithm 1 prefers using coarser-grained group-specific predictors higher up the tree, such as ALL or R1, while `Prepend` often resorts to the finer-grained predictors at the leaves.

Leaf	R1,M,Ya	R1,M,Ma	R1,M,Oa	R1,F,Ya	R1,F,Ma	R1,F,Oa
TREE	R1,M,Ya	R1,M,Ma	R1,M,Oa	R1,F,Ya	R1,F,Ma	R1,F,Oa
PREP	R1,M,Ya	R1,M,Ma	R1,M,Oa	R1,F,Ya	R1,F,Ma	R1,F,Oa
Leaf	R2,M,Ya	R2,M,Ma	R2,M,Oa	R2,F,Ya	R2,F,Ma	R2,F,Oa
TREE	R2,M,Ya	**ALL**	**ALL**	R2,F,Ya	R2,F,Ma	R2,F,Oa
PREP	R2,M,Ya	**R2,M,Ma**	**R2,M,Oa**	R2,F,Ya	R2,F,Ma	R2,F,Oa
Leaf	R3+,M,Ya	R3+,M,Ma	R3+,M,Oa	R3+,F,Ya	R3+,F,Ma	R3+,F,Oa
TREE	**ALL**	**ALL**	**ALL**	R3+,F,Ya	R3+,F,Ma	**ALL**
PREP	**R3+,M,Ya**	**R3+,M,Ma**	**R3+,M,Oa**	R3+,F,Ya	R3+,F,Ma	**R3+,F,Oa**
Leaf	R6+,M,Ya	R6+,M,Ma	R6+,M,Oa	R6+,F,Ya	R6+,F,Ma	R6+,F,Oa
TREE	R6+,M,Ya	R6+,M,Ma	R6+,M,Oa	R6+,F,Ya	R6+,F,Ma	R6+,F,Oa
PREP	R6+,M,Ya	R6+,M,Ma	R6+,M,Oa	R6+,F,Ya	R6+,F,Ma	R6+,F,Oa
Leaf	R7,M,Ya	R7,M,Ma	R7,M,Oa	R7,F,Ya	R7,F,Ma	R7,F,Oa
TREE	R7,M,Ya	R7,M,Ma	R7,M,Oa	R7,F,Ya	R7,F,Ma	**ALL**
PREP	R7,M,Ya	R7,M,Ma	R7,M,Oa	R7,F,Ya	R7,F,Ma	**R7,F,Oa**
Leaf	R8,M,Ya	R8,M,Ma	R8,M,Oa	R8,F,Ya	R8,F,Ma	R8,F,Oa
TREE	R8,M,Ya	**R8,M**	**R8,M**	**ALL**	R8,F,Ma	R8,F,Oa
PREP	R8,M,Ya	**R8,M,Ma**	**R8,M,Oa**	**R8,F,Ya**	R8,F,Ma	R8,F,Oa

Table 4. Predictors for each `race-sex-age` leaf node in CA Employment dataset. ALL indicates the ERM logistic regression predictor trained on all the data. Bolded entries are leaves where `MGL−Tree` and `Prepend` differ in their predictor.
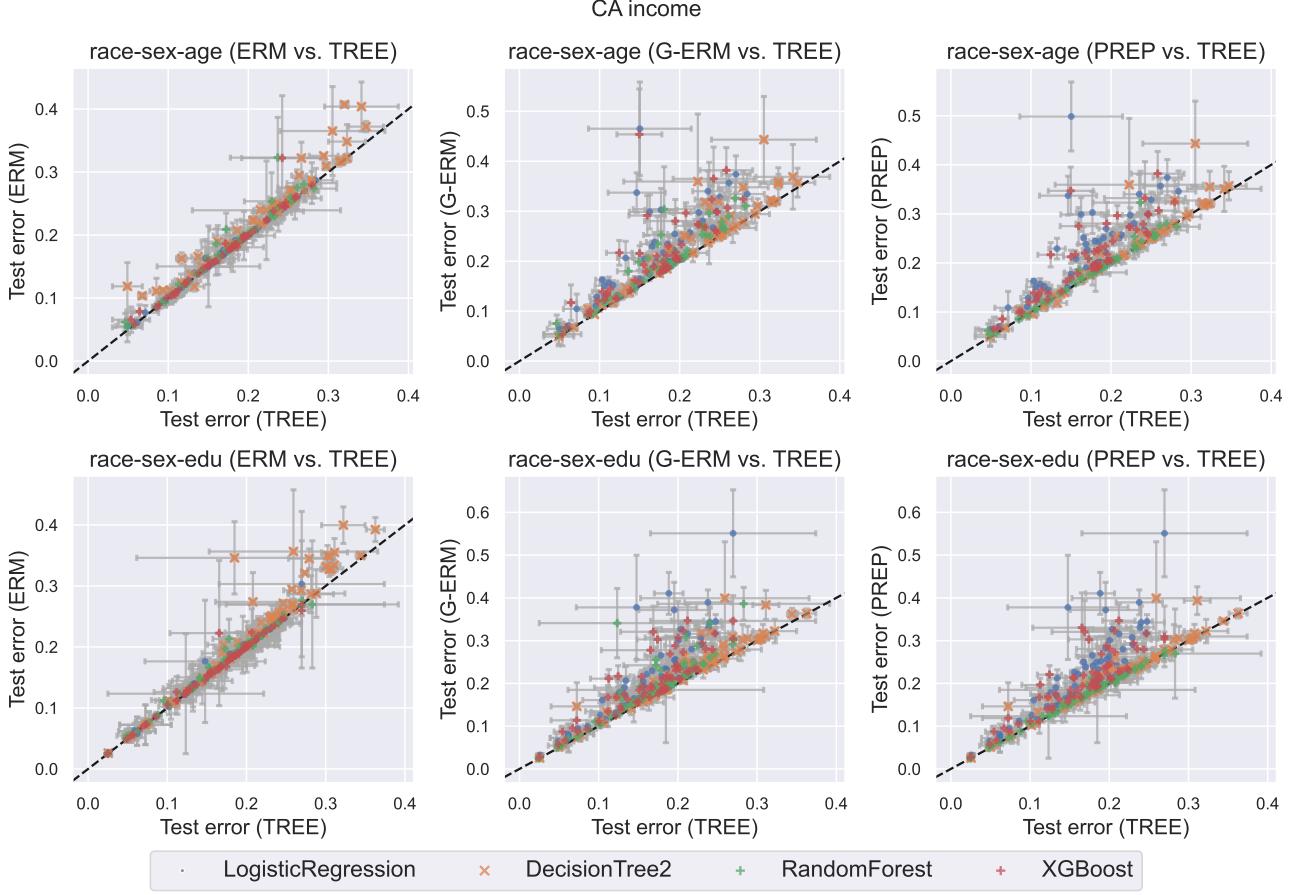
Leaf	R1,M,Ya	R1,M,Ma	R1,M,Oa	R1,F,Ya	R1,F,Ma	R1,F,Oa
TREE	**R1**	**R1**	**R1**	**R1**	**R1**	R1,F,Oa
PREP	**R1,M,Ya**	**R1,M,Ma**	**R1,M,Oa**	**R1,F,Ya**	**R1,F,Ma**	R1,F,Oa
Leaf	R2,M,Ya	R2,M,Ma	R2,M,Oa	R2,F,Ya	R2,F,Ma	R2,F,Oa
TREE	**ALL**	**ALL**	**ALL**	**ALL**	ALL	**ALL**
PREP	**R2,M,Ya**	**R2,M,Ma**	**R2,M,Oa**	**R2,F,Ya**	ALL	**R2,F,Oa**
Leaf	R3+,M,Ya	R3+,M,Ma	R3+,M,Oa	R3+,F,Ya	R3+,F,Ma	R3+,F,Oa
TREE	**ALL**	ALL	**ALL**	**ALL**	**ALL**	**ALL**
PREP	**R3+,M,Ya**	ALL	**R3+,M,Oa**	**R3+,F,Ya**	**R3+,F,Ma**	**R3+,F,Oa**
Leaf	R6+,M,Ya	R6+,M,Ma	R6+,M,Oa	R6+,F,Ya	R6+,F,Ma	R6+,F,Oa
TREE	**R6+**	**R6+,M,Ma**	**R6+**	**R6+**	**R6+**	**R6+**
PREP	ALL	ALL	ALL	ALL	**R6+,F,Ma**	**R6+,F,Oa**
Leaf	R7,M,Ya	R7,M,Ma	R7,M,Oa	R7,F,Ya	R7,F,Ma	R7,F,Oa
TREE	R7,M,Ya	R7,M,Ma	R7,M,Oa	R7,F,Ya	**R7,F,Ma**	ALL
PREP	R7,M,Ya	R7,M,Ma	R7,M,Oa	R7,F,Ya	**ALL**	ALL
Leaf	R8,M,Ya	R8,M,Ma	R8,M,Oa	R8,F,Ya	R8,F,Ma	R8,F,Oa
TREE	R8,M,Ya	**ALL**	**ALL**	**ALL**	R8,F,Ma	R8,F,Oa
PREP	R8,M,Ya	**R8,M,Ma**	**R8,M,Oa**	**R8,F,Ya**	R8,F,Ma	R8,F,Oa

Table 5. Predictors for each `race-sex-age` leaf node in CA Income dataset. `ALL` indicates the ERM logistic regression predictor trained on all the data. Bolded entries are leaves where `MGL-Tree` and `Prepend` differ in their predictor.

H. Additional experimental results

In this section, we provide additional results for the other datasets not shown in the main body. For each dataset, we consider the two collections of hierarchical groups described in Appendix F, and we compare the generalization performance of all the methods and benchmark hypothesis classes described in Section 4.1.

For all the datasets, we plot the test error of each method on a held-out test set for each group. Each group corresponds to a point on the plot. The $y = x$ line corresponds to equal test error between the methods; all points above the line indicates that our Algorithm 1 had lower test error. Error bars are from the standard error from 10 random trials on each group (from a fresh train-test split, retraining each model from scratch). Error bars are omitted for the nationwide datasets for visual presentation.

Throughout all the figures, "DecisionTree2," "DecisionTree4," and "DecisionTree8" refer to decision trees trained with the `max_depth` parameter set to 2, 4, and 8, respectively. "ERM" refers to the model trained on all available data, "G-ERM" refers to the group-stratified model trained only on data from the group corresponding to the point on the plot, and "PREP" refers to the `Prepend` algorithm of Tosh & Hsu (2022).

We also include a "case study" of a particular dataset's (CA Employment) error rates group-by-group for logistic regression. We see that the bars for Algorithm 1 are consistently on par with or lower than both the ERM and Group ERM bars, sometimes by a very significant margin. In cases where Algorithm 1 beats ERM by a significant amount (such as on, say, the group `R1, M, Oa`), the predictor from Algorithm 1 has identified an instance of a high-error subgroup on a specific slice of the population, which may prove useful for further auditing.

CA income

Figure 6. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (CA Income). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: logistic regression, decision trees with `max_depth` 2, random forest, and XGBoost.
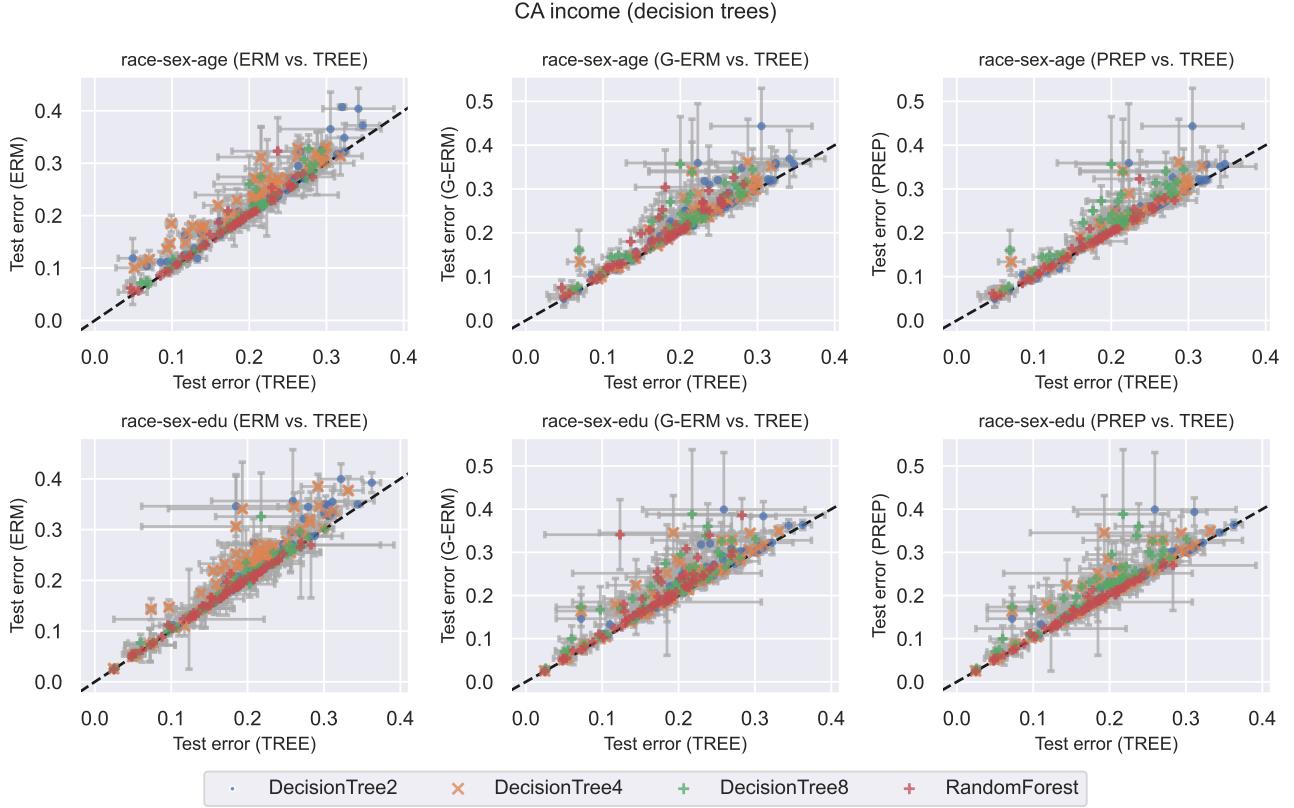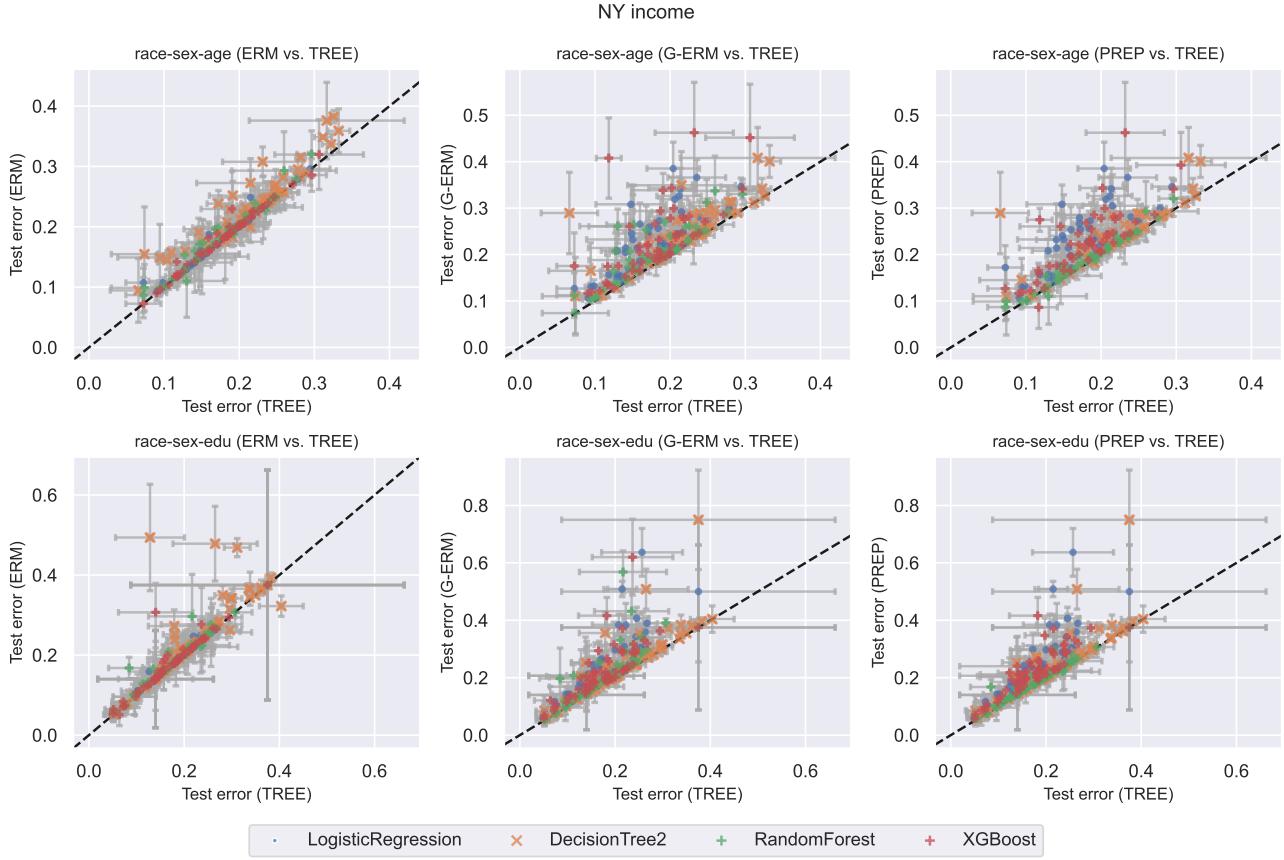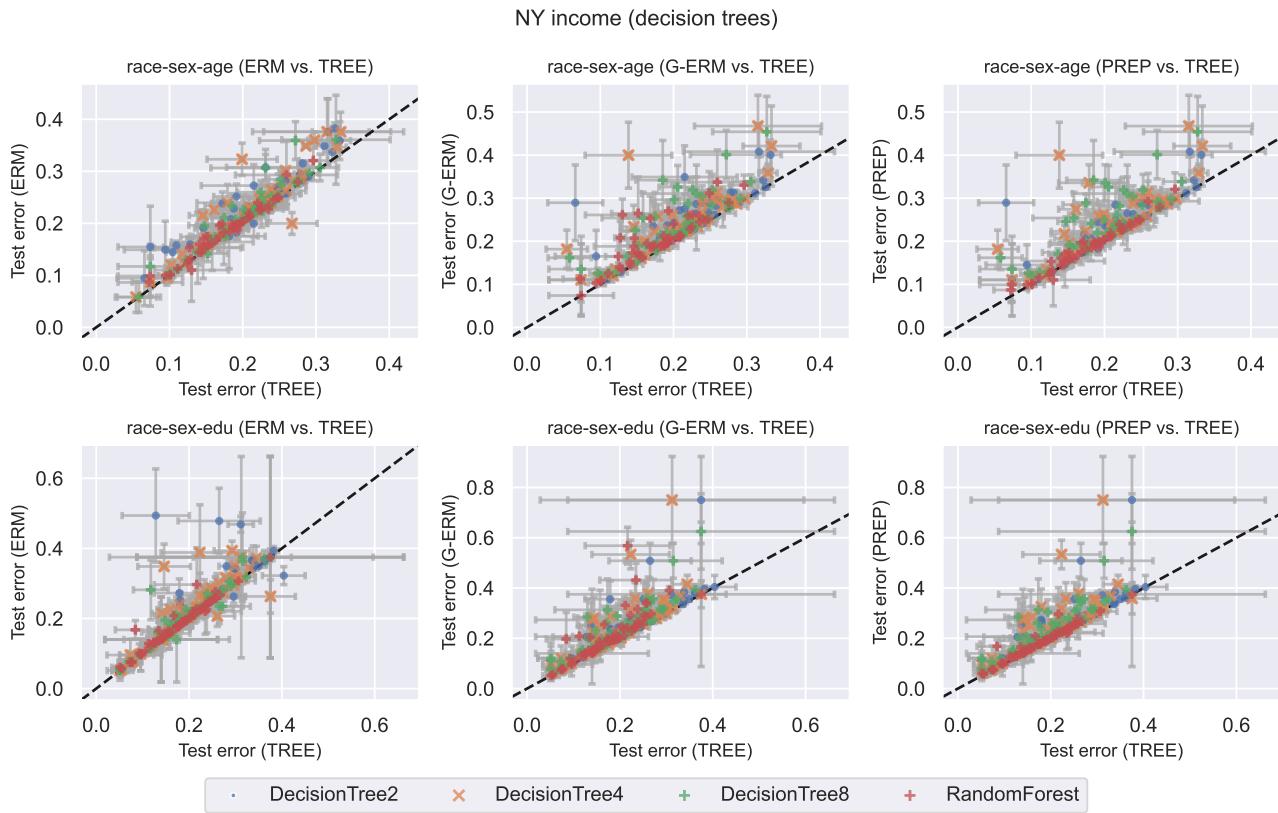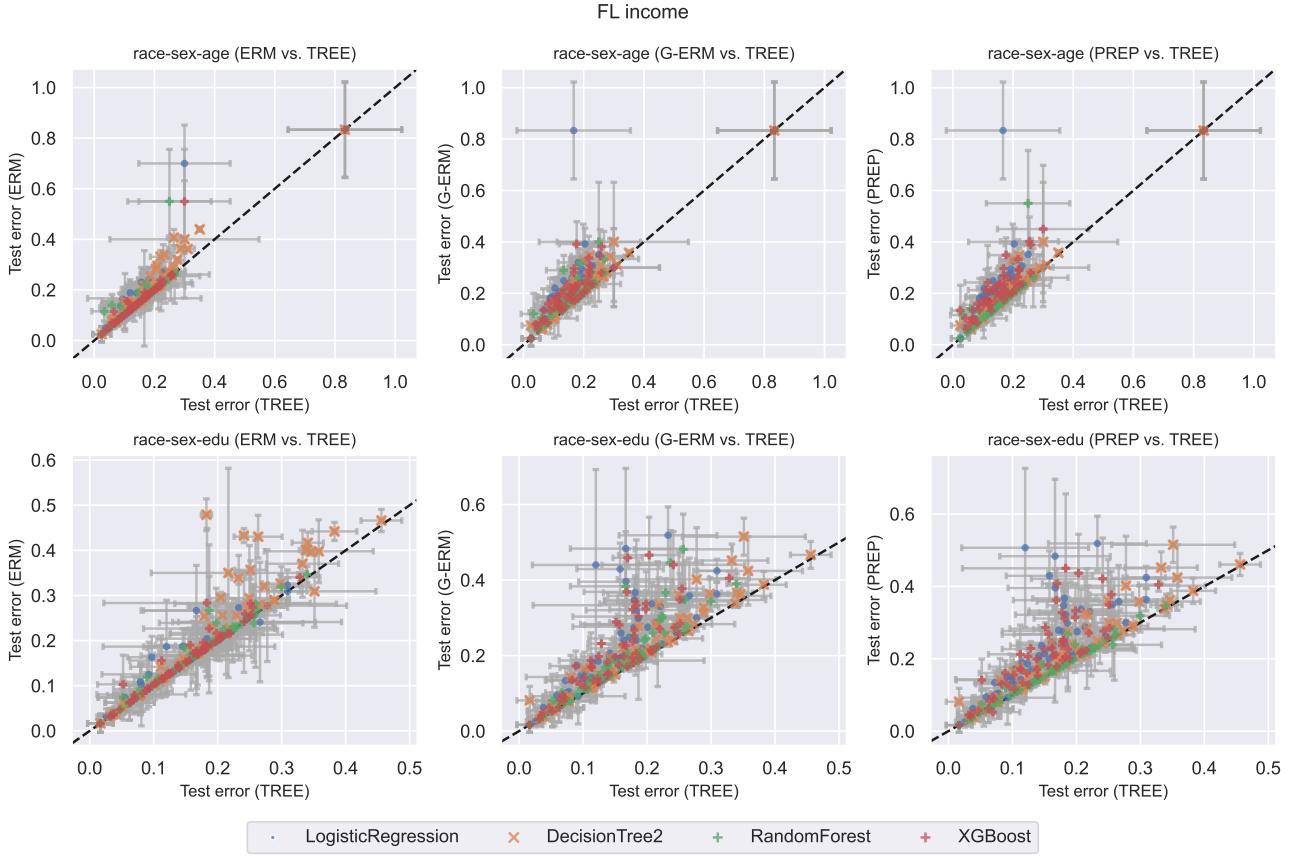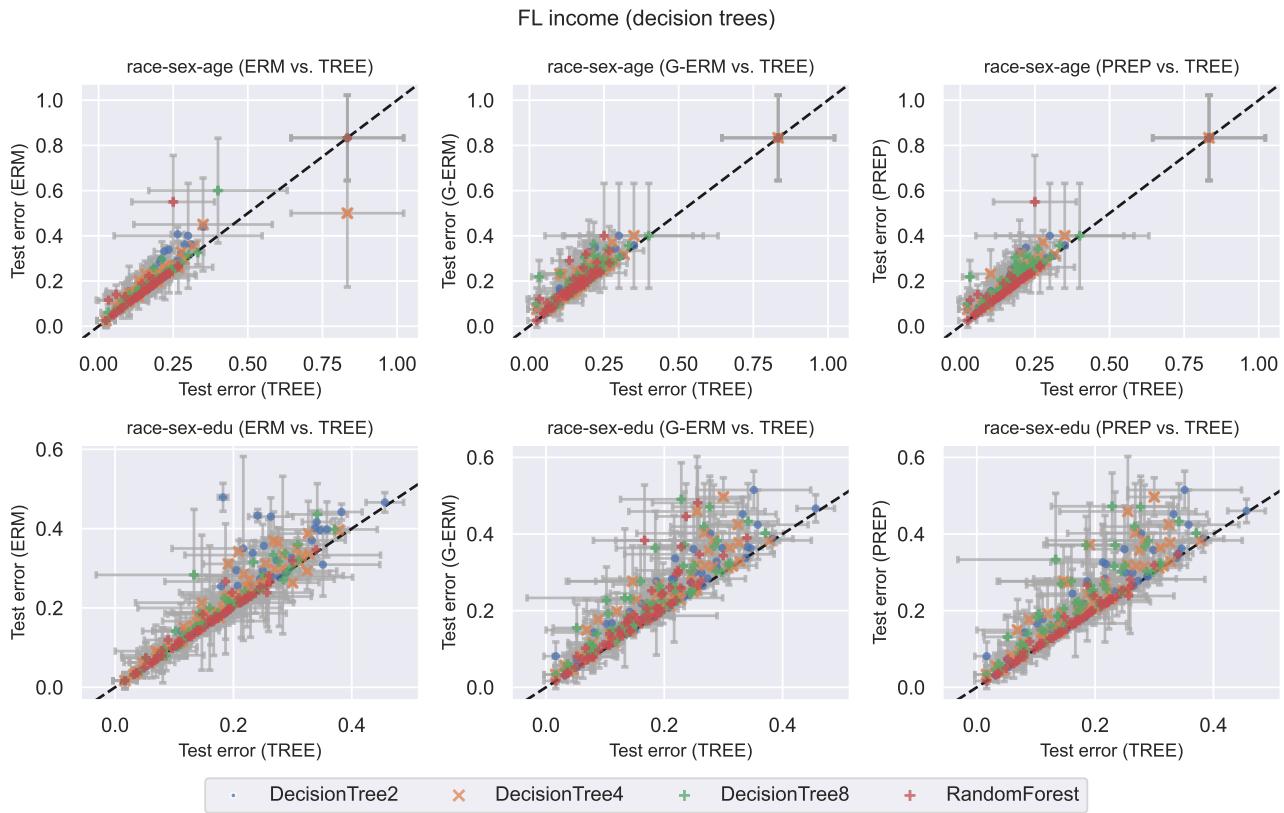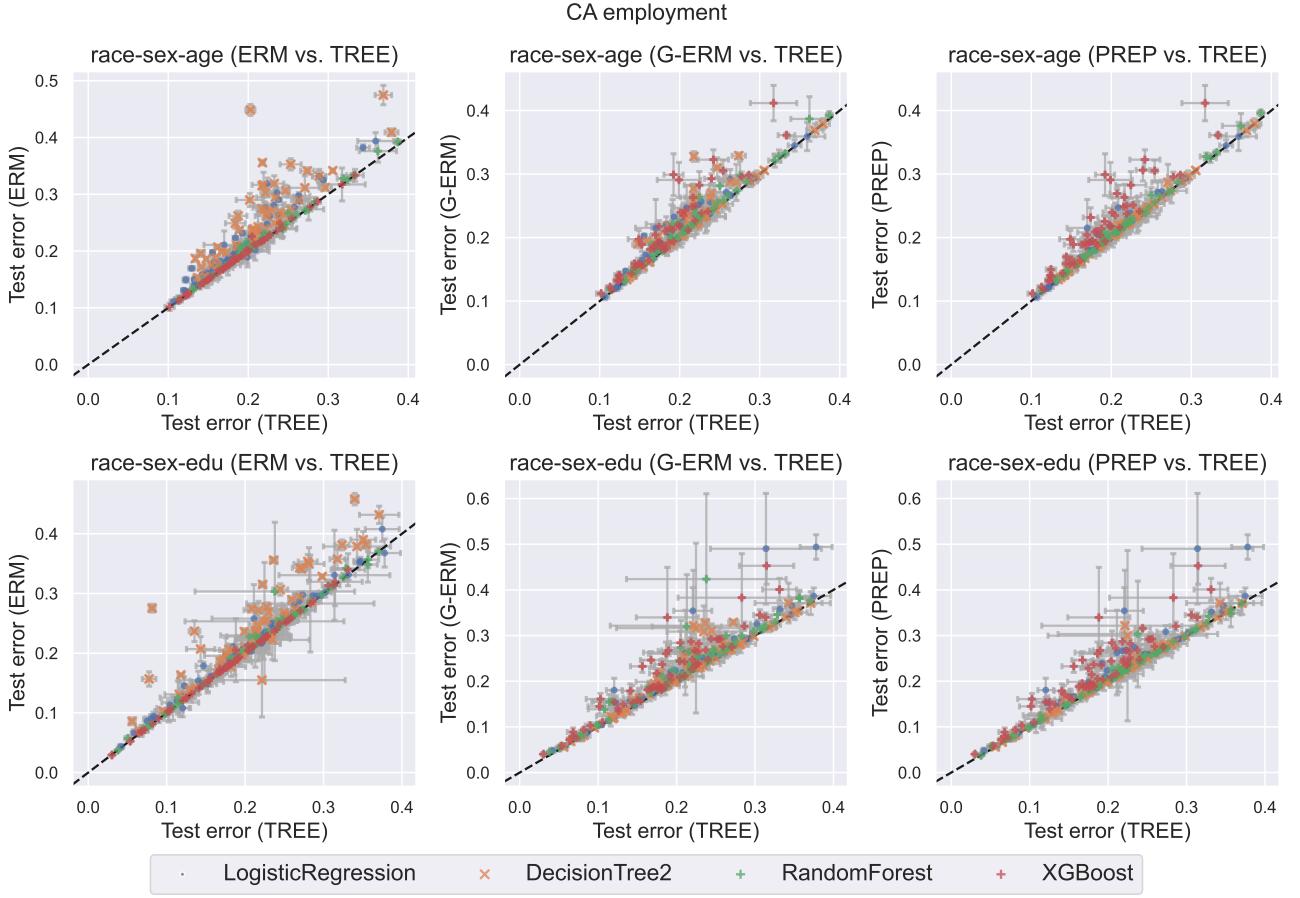
Figure 7. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (CA Income). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: decision trees with `max_depth` 2, decision trees with `max_depth` 4, decision trees with `max_depth` 8, and random forest.
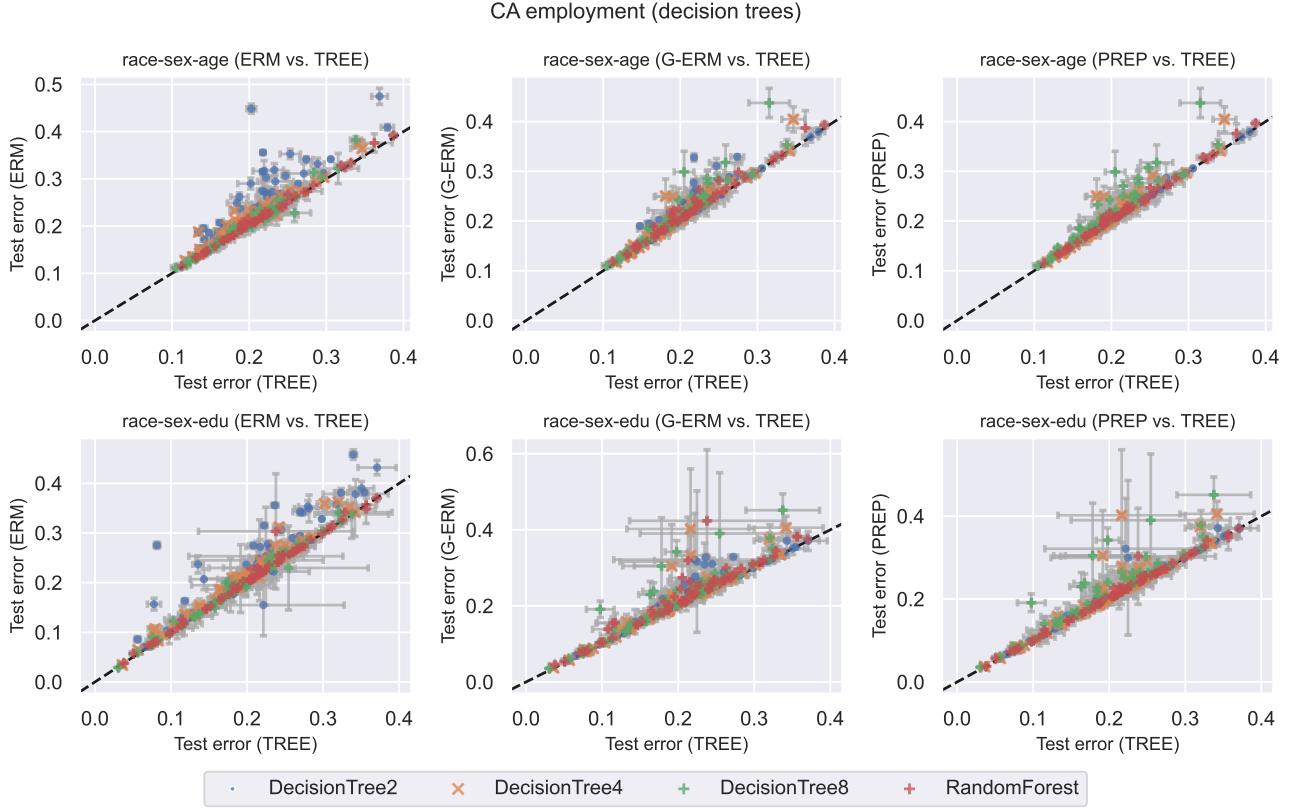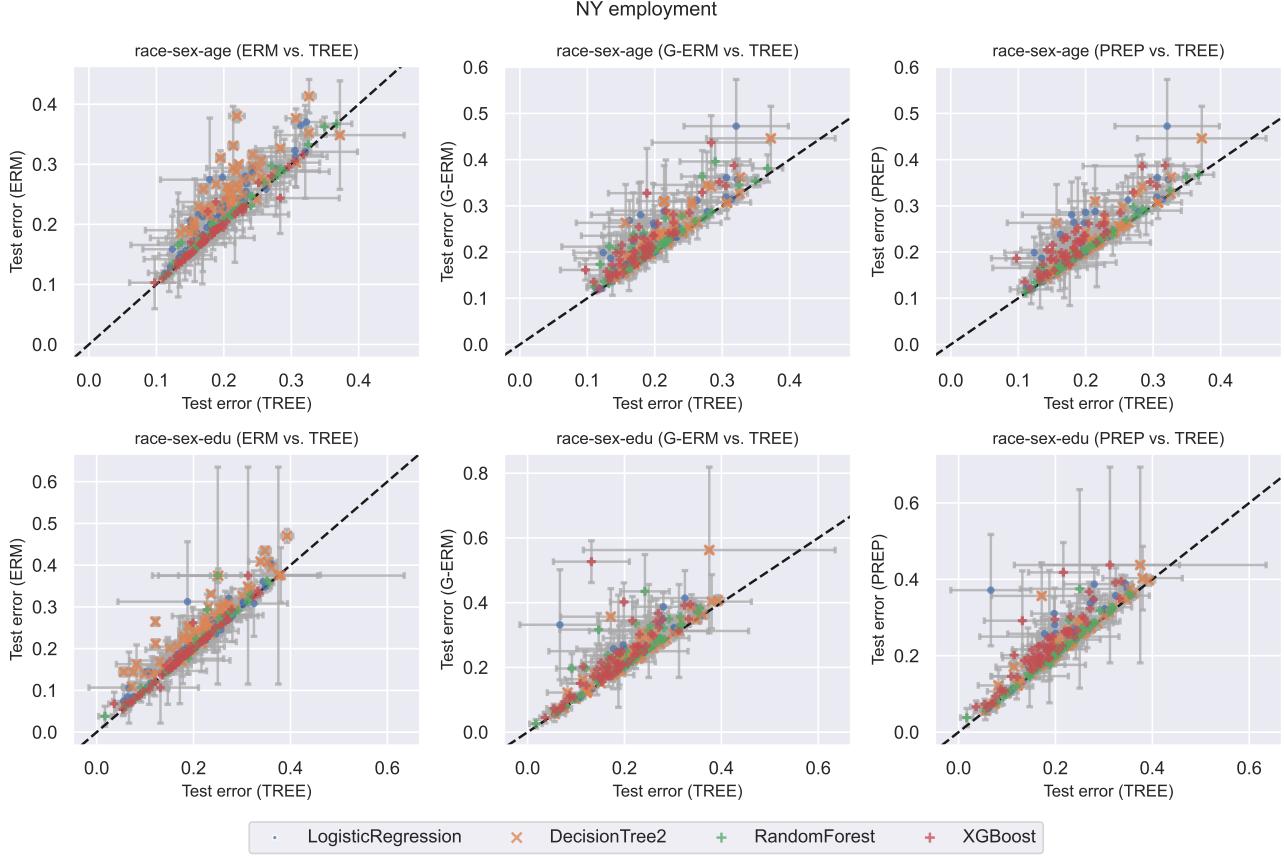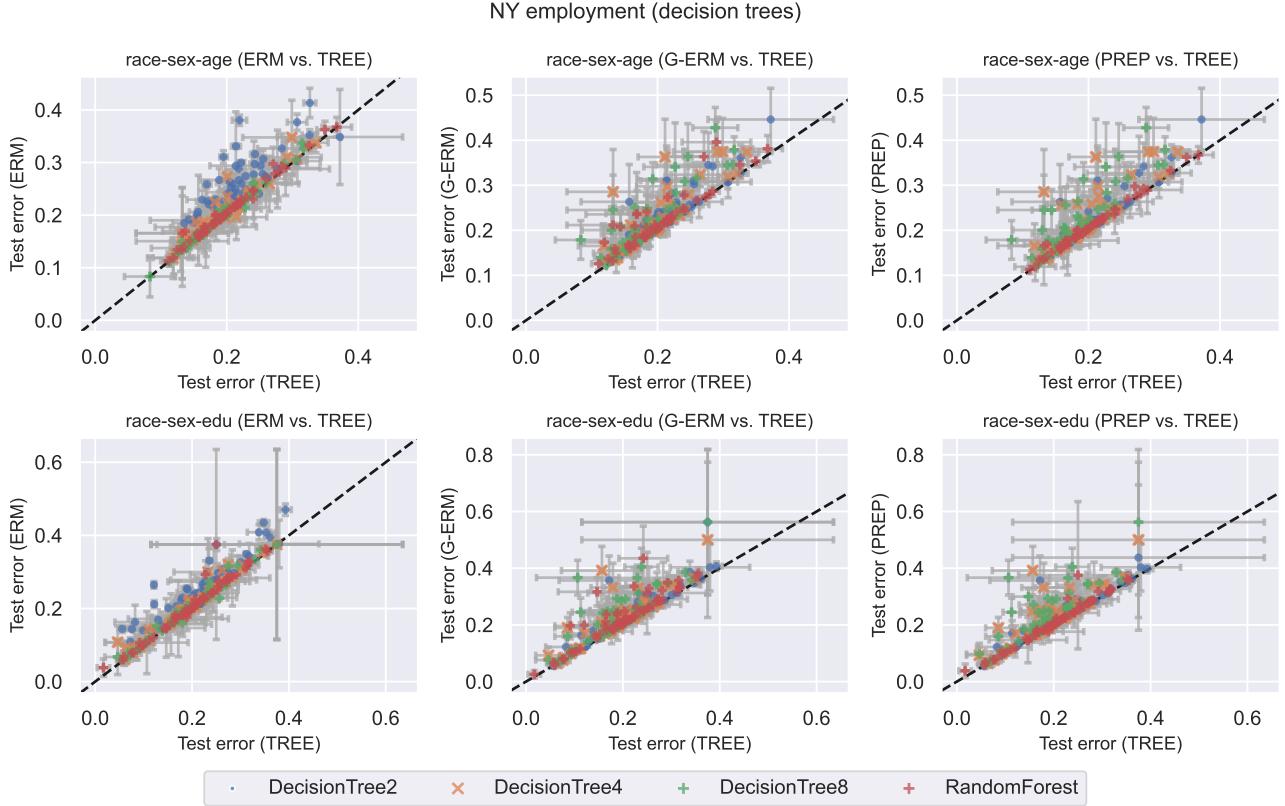
Figure 8. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (NY Income). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: logistic regression, decision trees with `max_depth` 2, random forest, and XGBoost.

Figure 9. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (NY Income). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: decision trees with `max_depth` 2, decision trees with `max_depth` 4, decision trees with `max_depth` 8, and random forest.

Figure 10. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (FL Income). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: logistic regression, decision trees with `max_depth` 2, random forest, and XGBoost.

Figure 11. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (FL Income). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: decision trees with `max_depth` 2, decision trees with `max_depth` 4, decision trees with `max_depth` 8, and random forest.
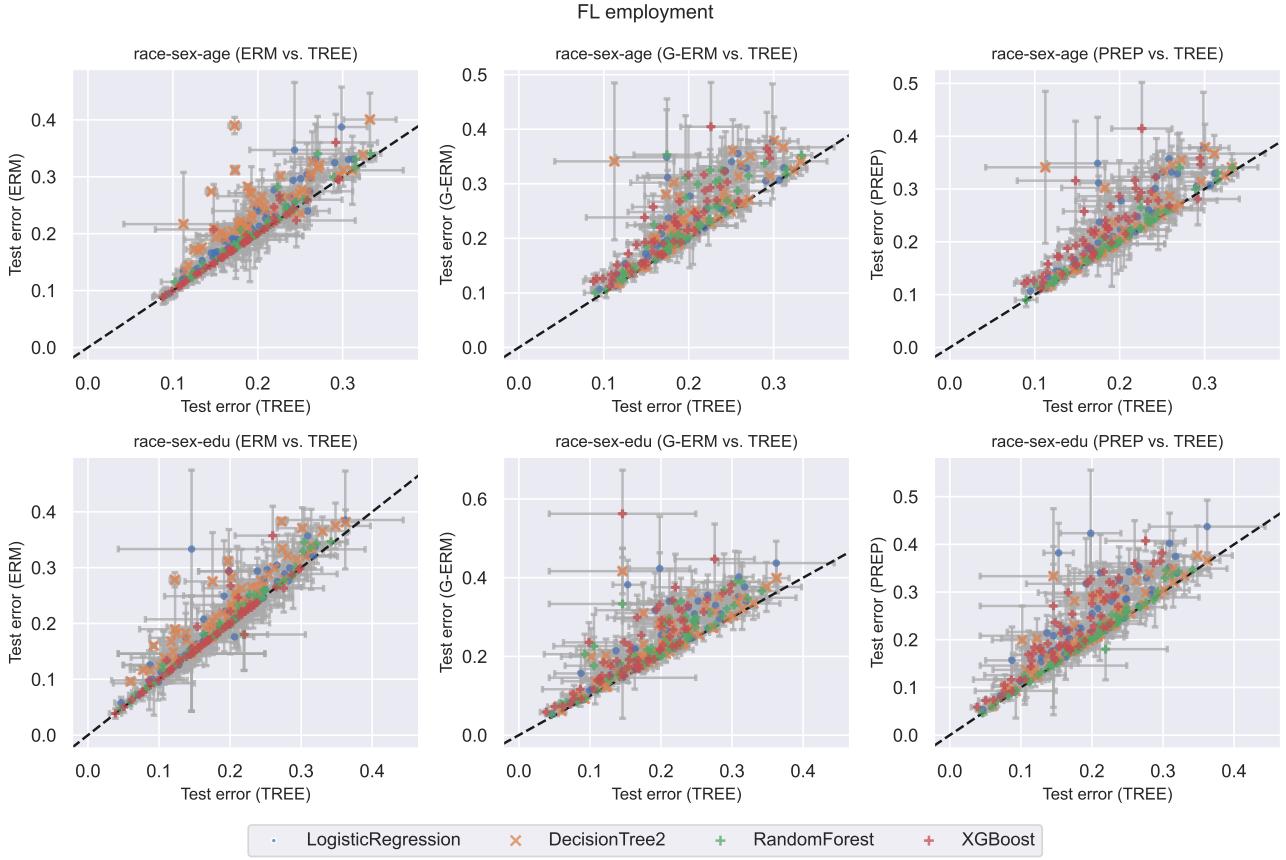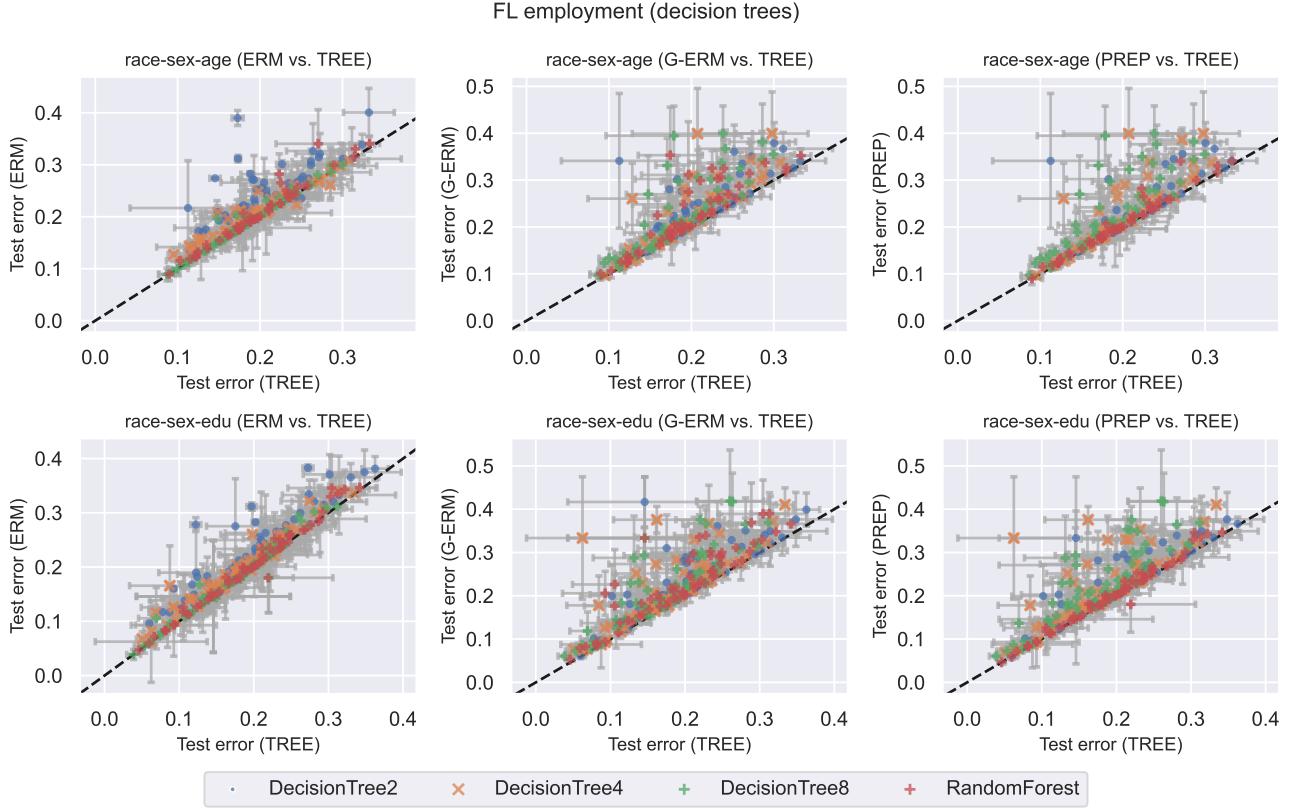
Figure 12. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (CA Employment). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: logistic regression, decision trees with `max_depth` 2, random forest, and XGBoost.

Figure 13. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (CA Employment). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: decision trees with `max_depth` 2, decision trees with `max_depth` 4, decision trees with `max_depth` 8, and random forest.
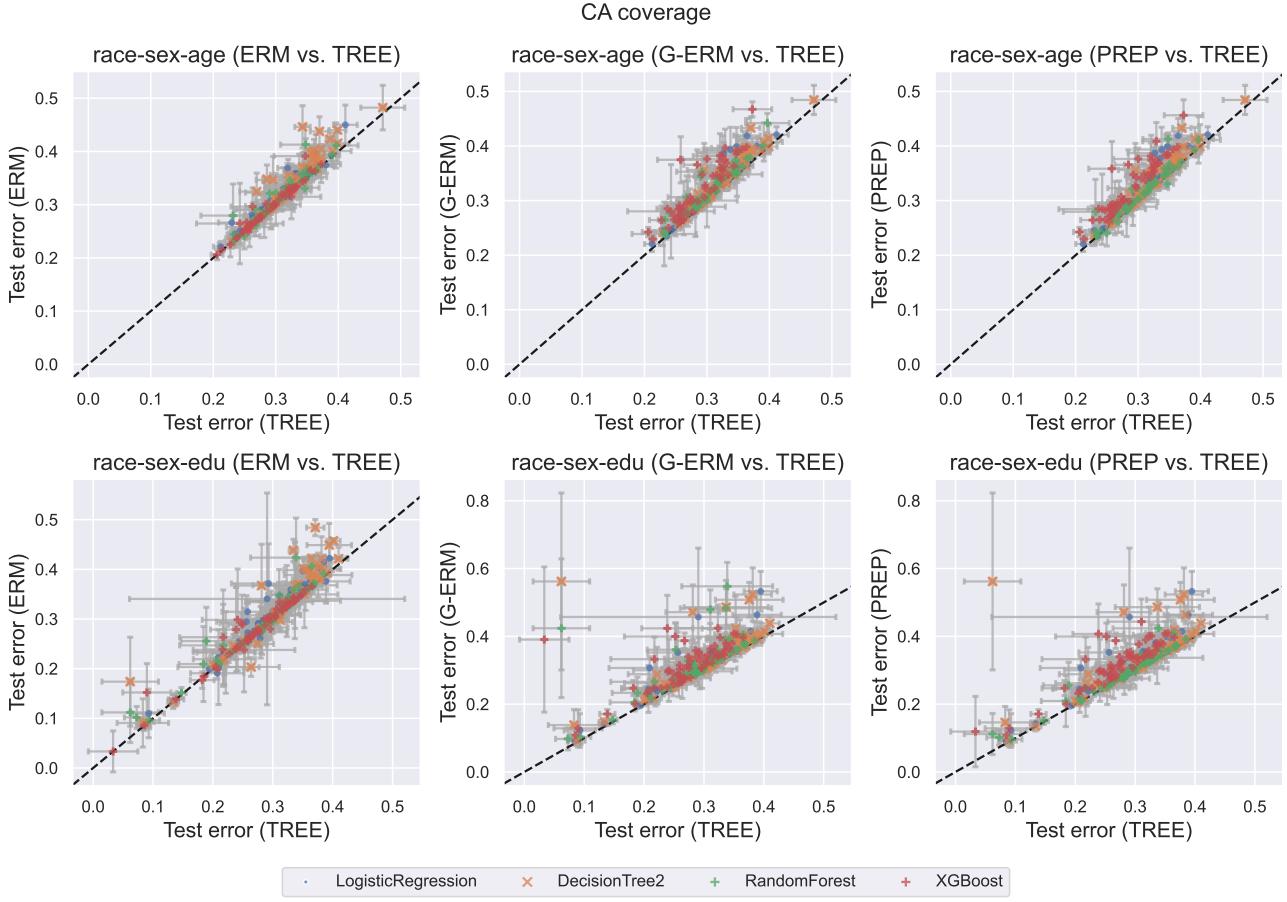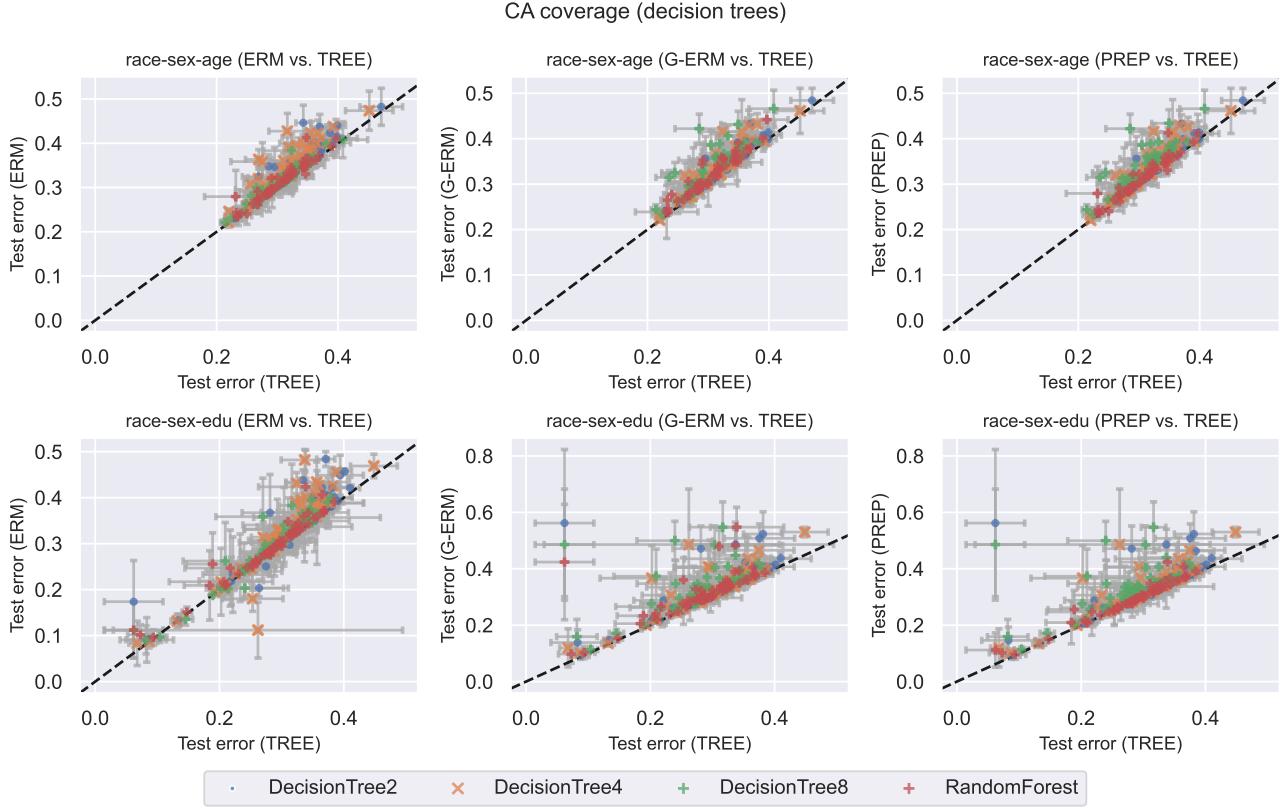
Figure 14. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (NY Employment). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: logistic regression, decision trees with `max_depth` 2, random forest, and XGBoost.

Figure 15. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (NY Employment). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: decision trees with `max_depth` 2, decision trees with `max_depth` 4, decision trees with `max_depth` 8, and random forest.
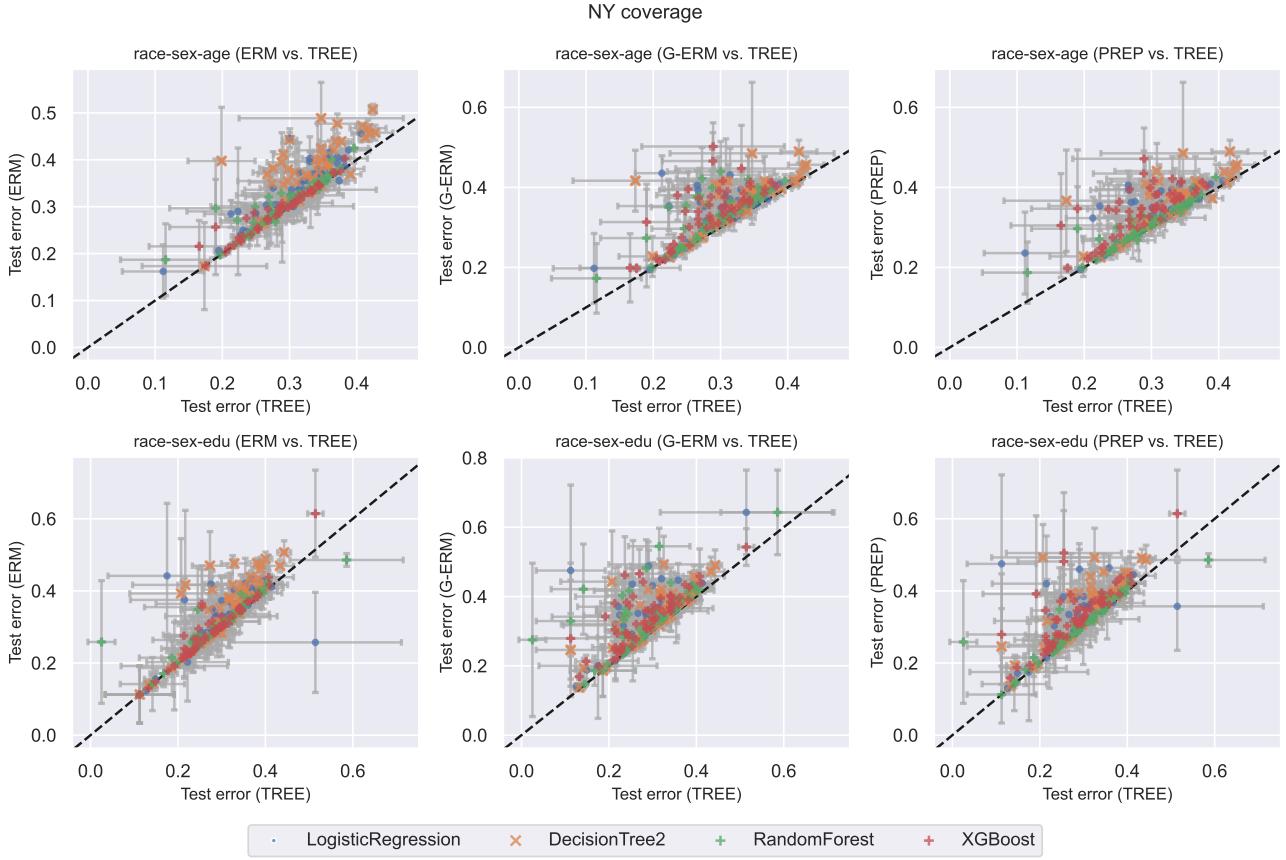
Figure 16. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (FL Employment). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: logistic regression, decision trees with `max_depth` 2, random forest, and XGBoost.

Figure 17. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (FL Employment). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: decision trees with `max_depth` 2, decision trees with `max_depth` 4, decision trees with `max_depth` 8, and random forest.

Figure 18. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (CA Coverage). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: logistic regression, decision trees with `max_depth` 2, random forest, and XGBoost.
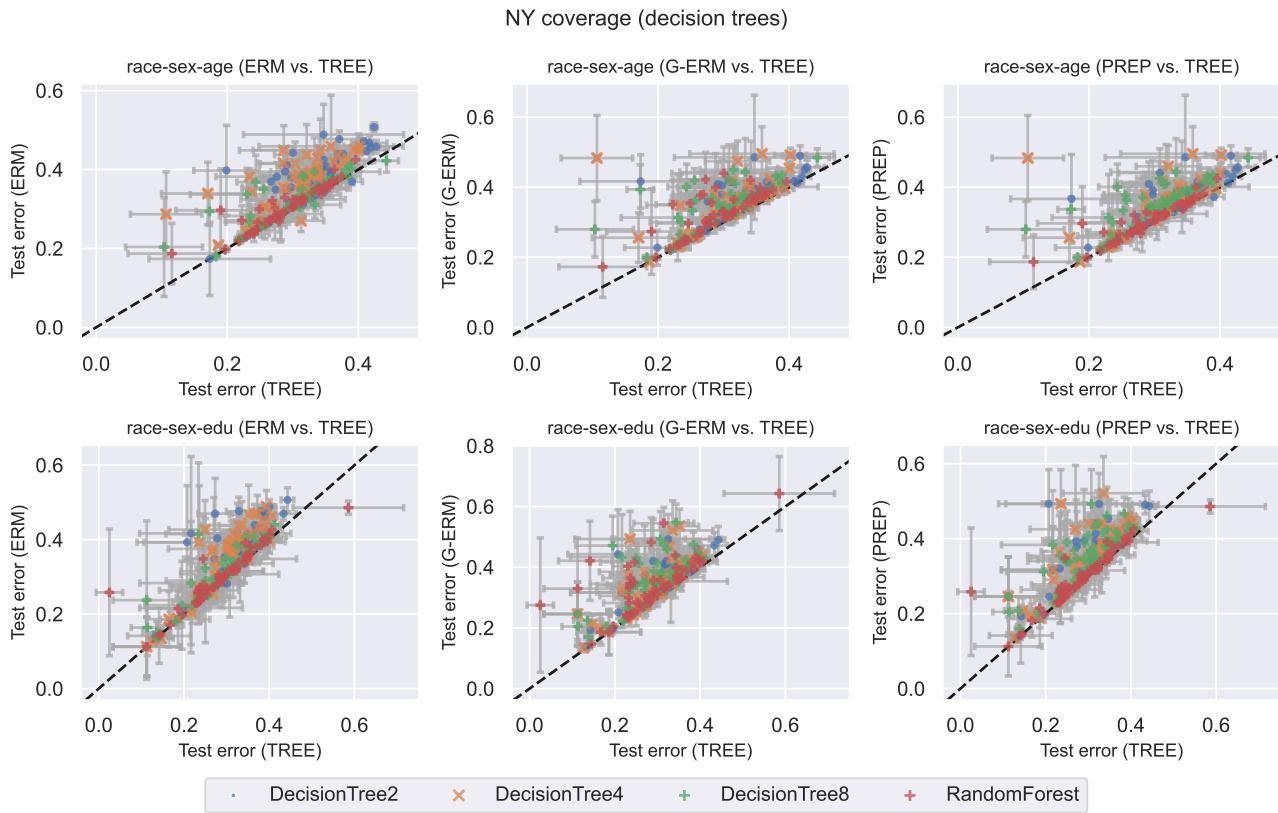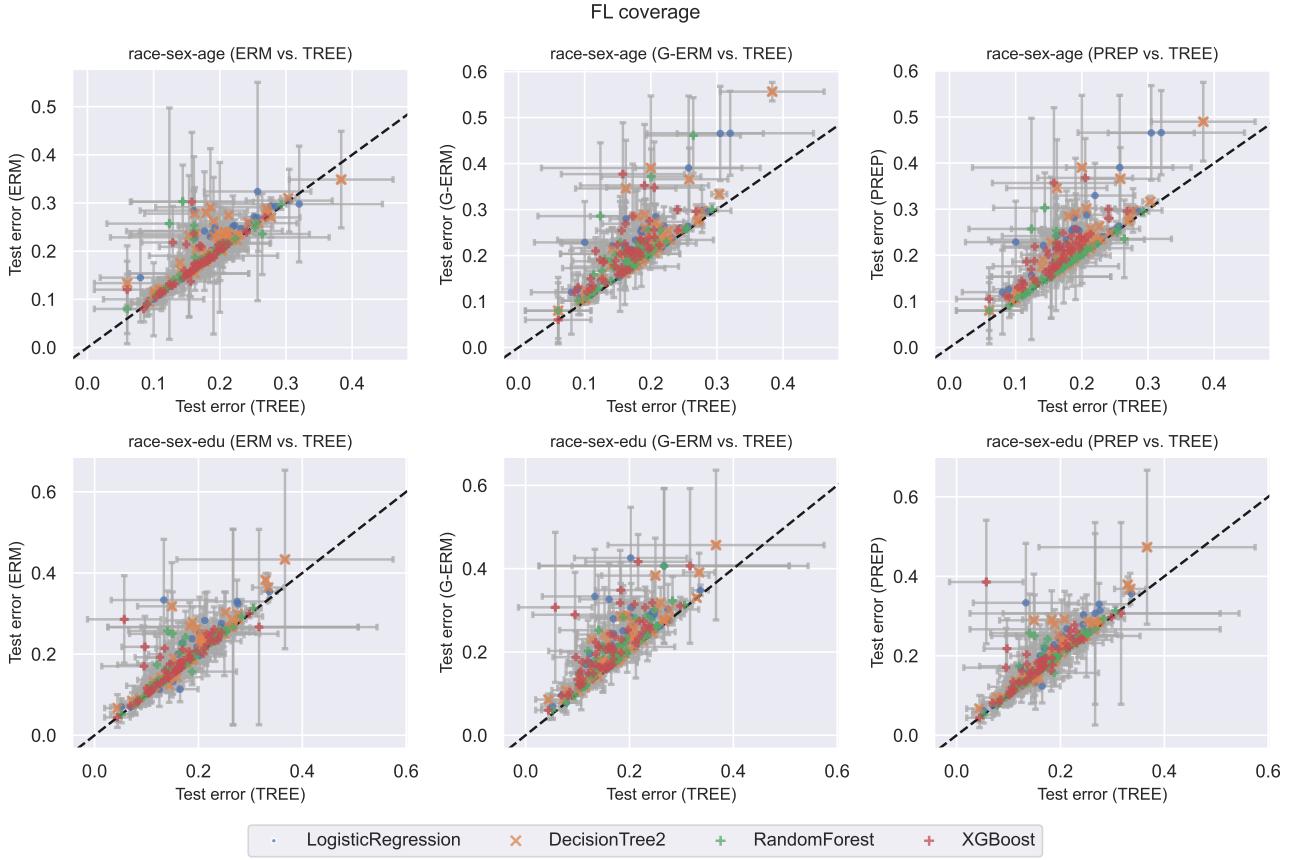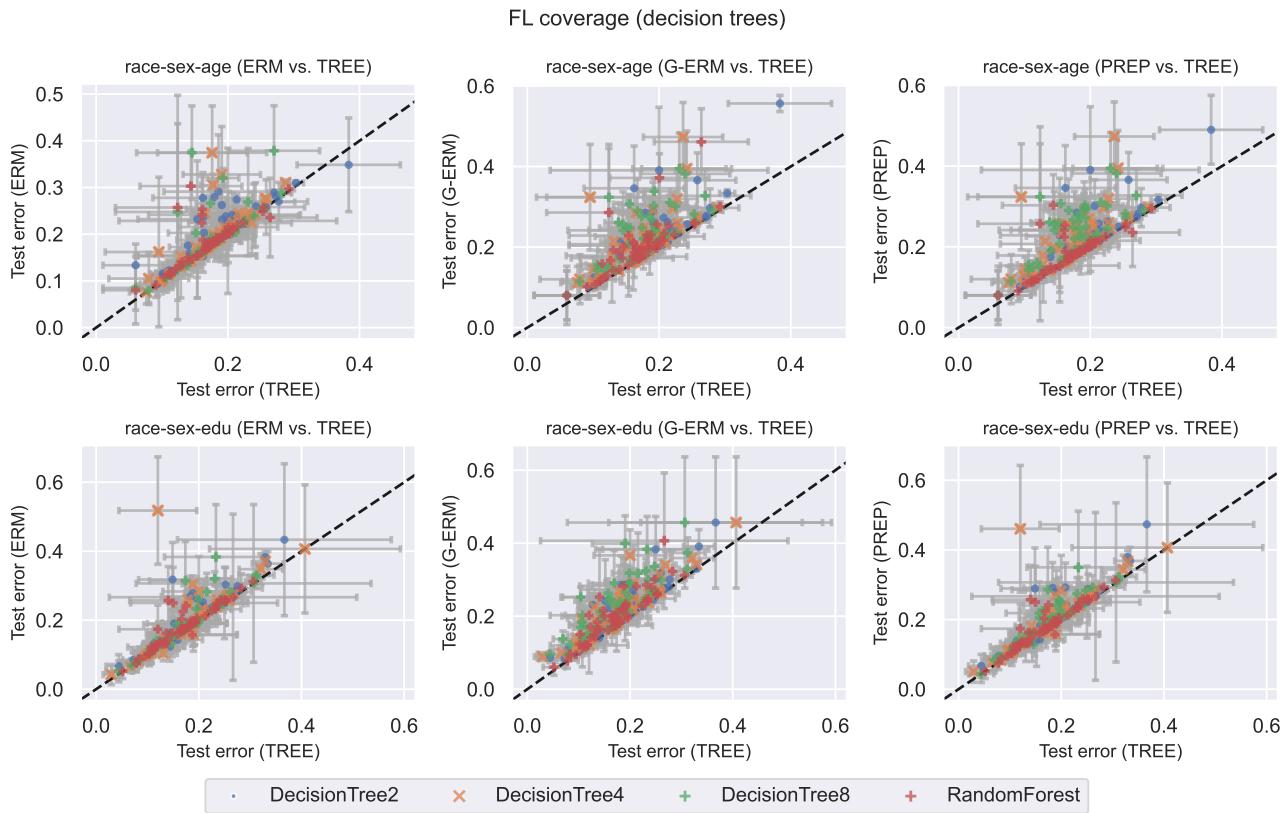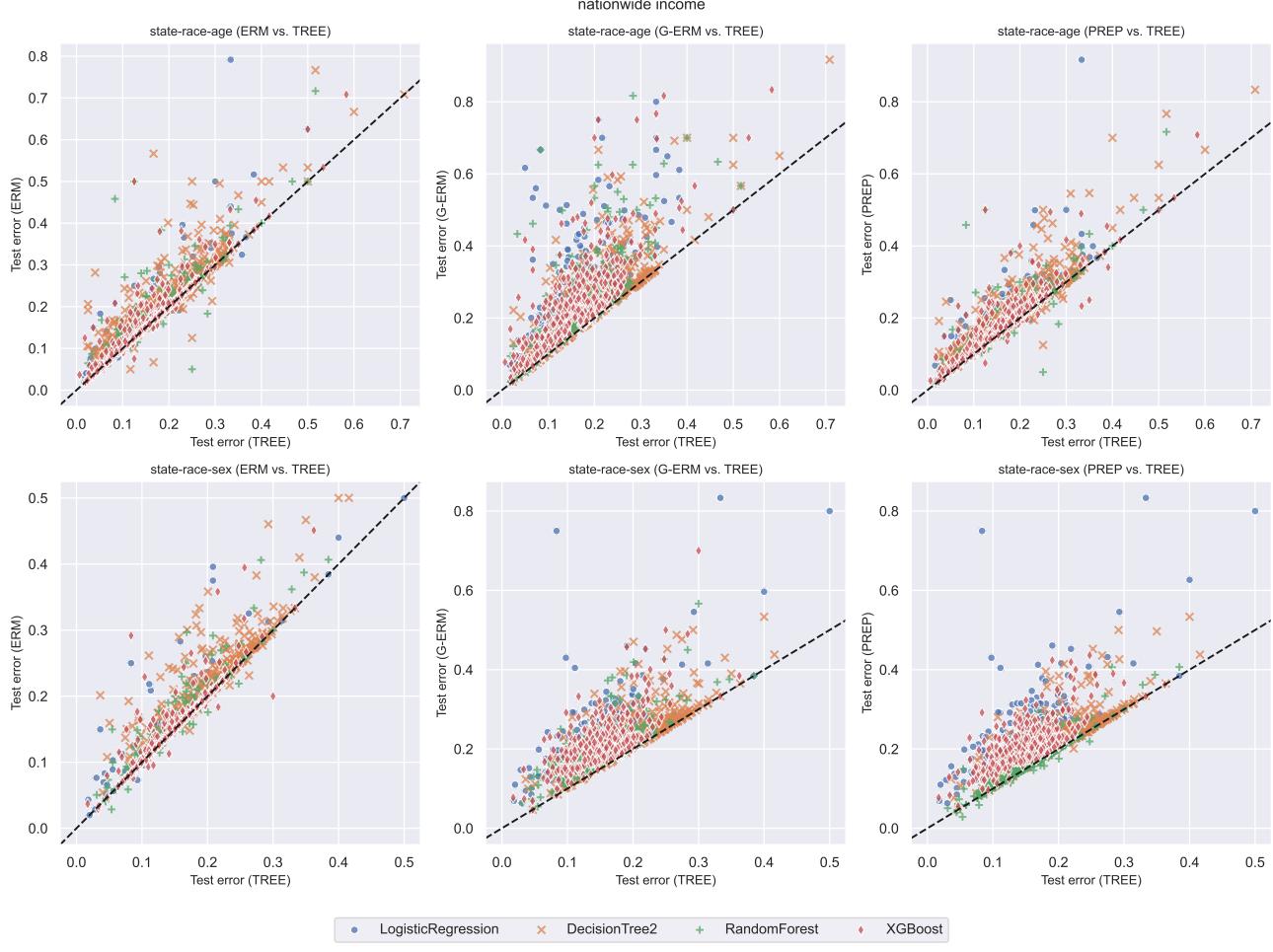
Figure 19. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (CA Coverage). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: decision trees with `max_depth` 2, decision trees with `max_depth` 4, decision trees with `max_depth` 8, and random forest.
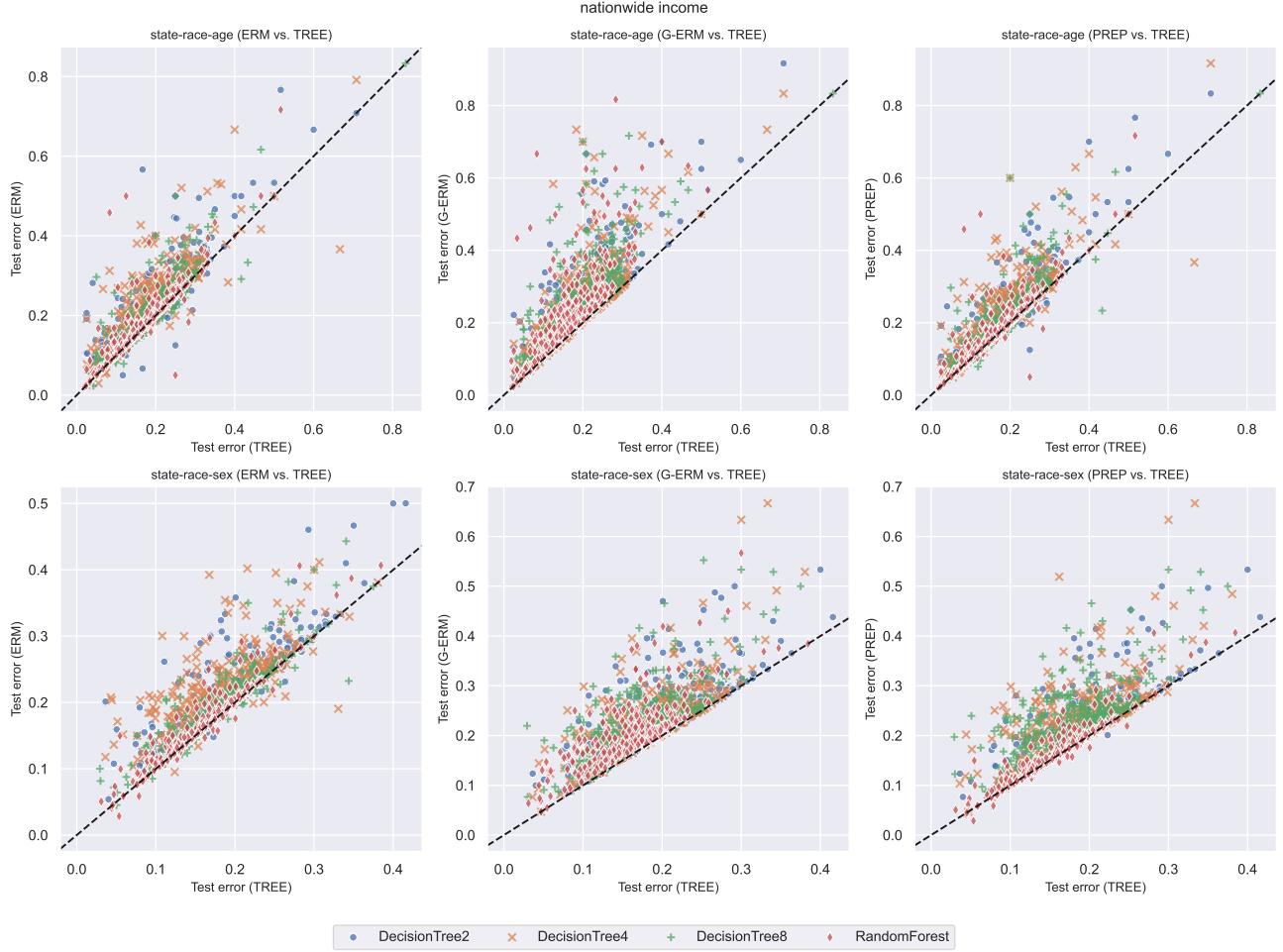
Figure 20. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (NY Coverage). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: logistic regression, decision trees with `max_depth` 2, random forest, and XGBoost.
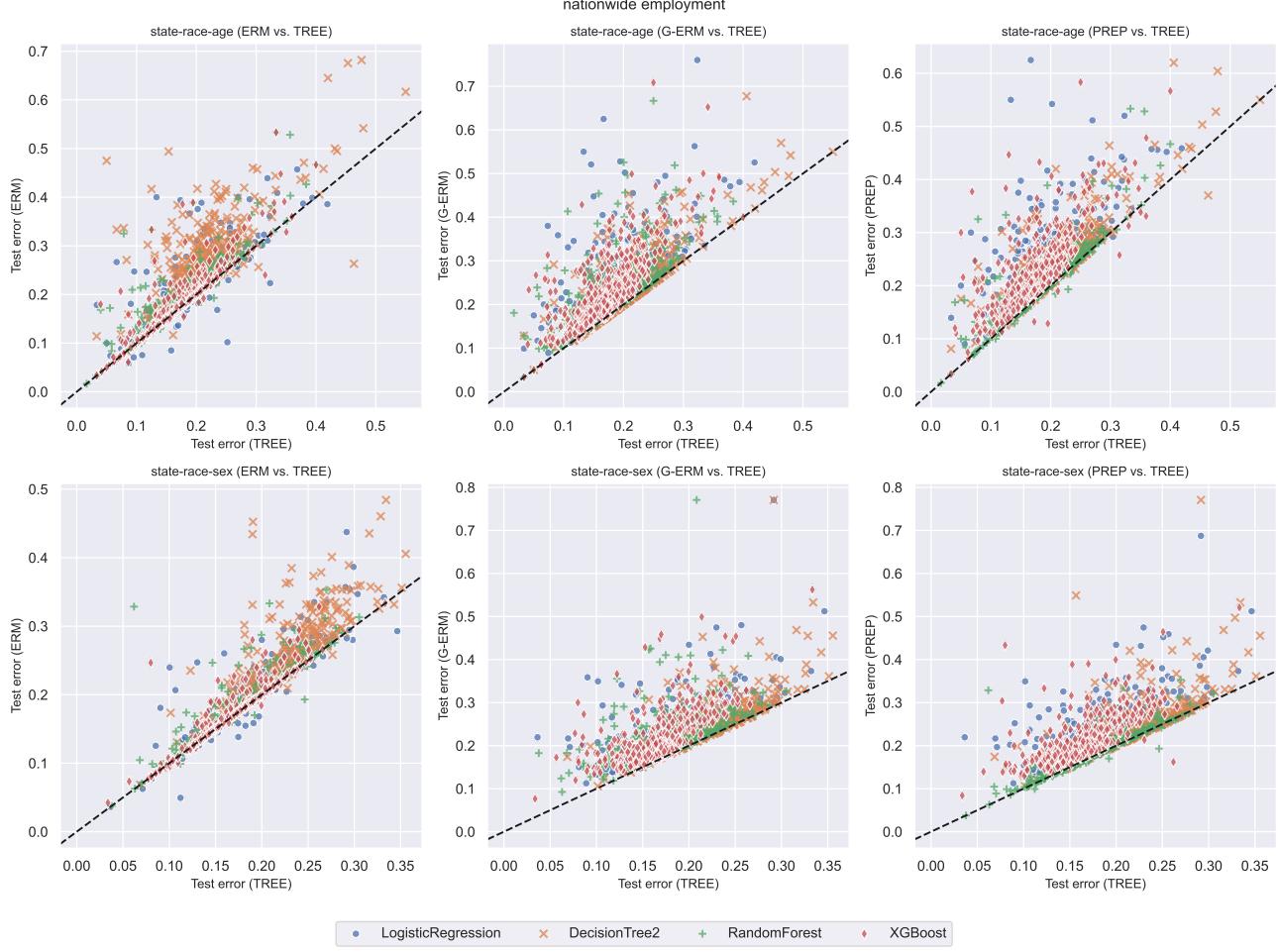
Figure 21. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (NY Coverage). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: decision trees with `max_depth` 2, decision trees with `max_depth` 4, decision trees with `max_depth` 8, and random forest.

Figure 22. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (FL Coverage). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: logistic regression, decision trees with `max_depth` 2, random forest, and XGBoost.

Figure 23. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups (FL Coverage). Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: decision trees with `max_depth` 2, decision trees with `max_depth` 4, decision trees with `max_depth` 8, and random forest.
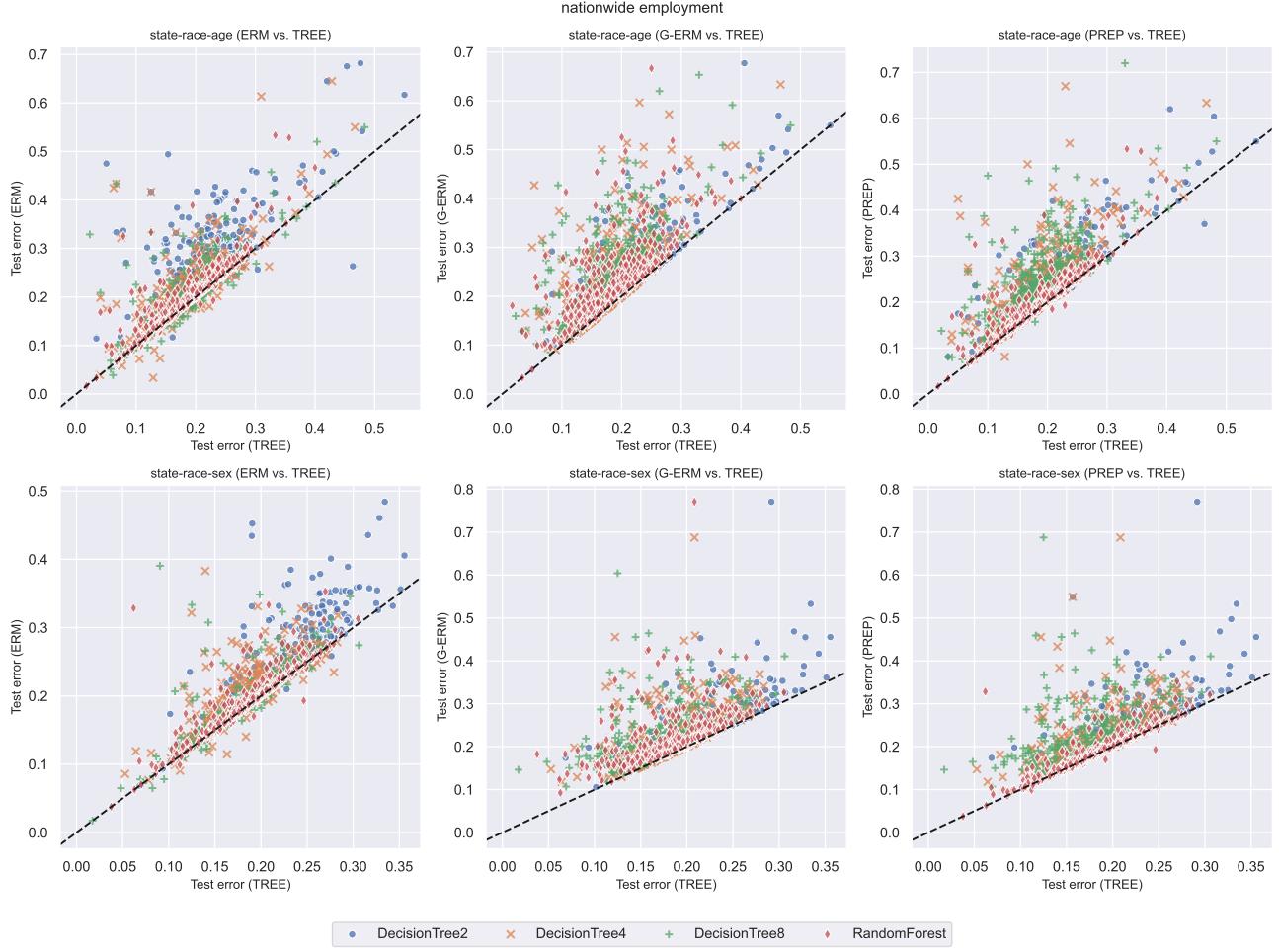
Figure 24. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups on nationwide dataset for the Income task. Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: logistic regression, decision trees with `max_depth` 2, random forest, and XGBoost.
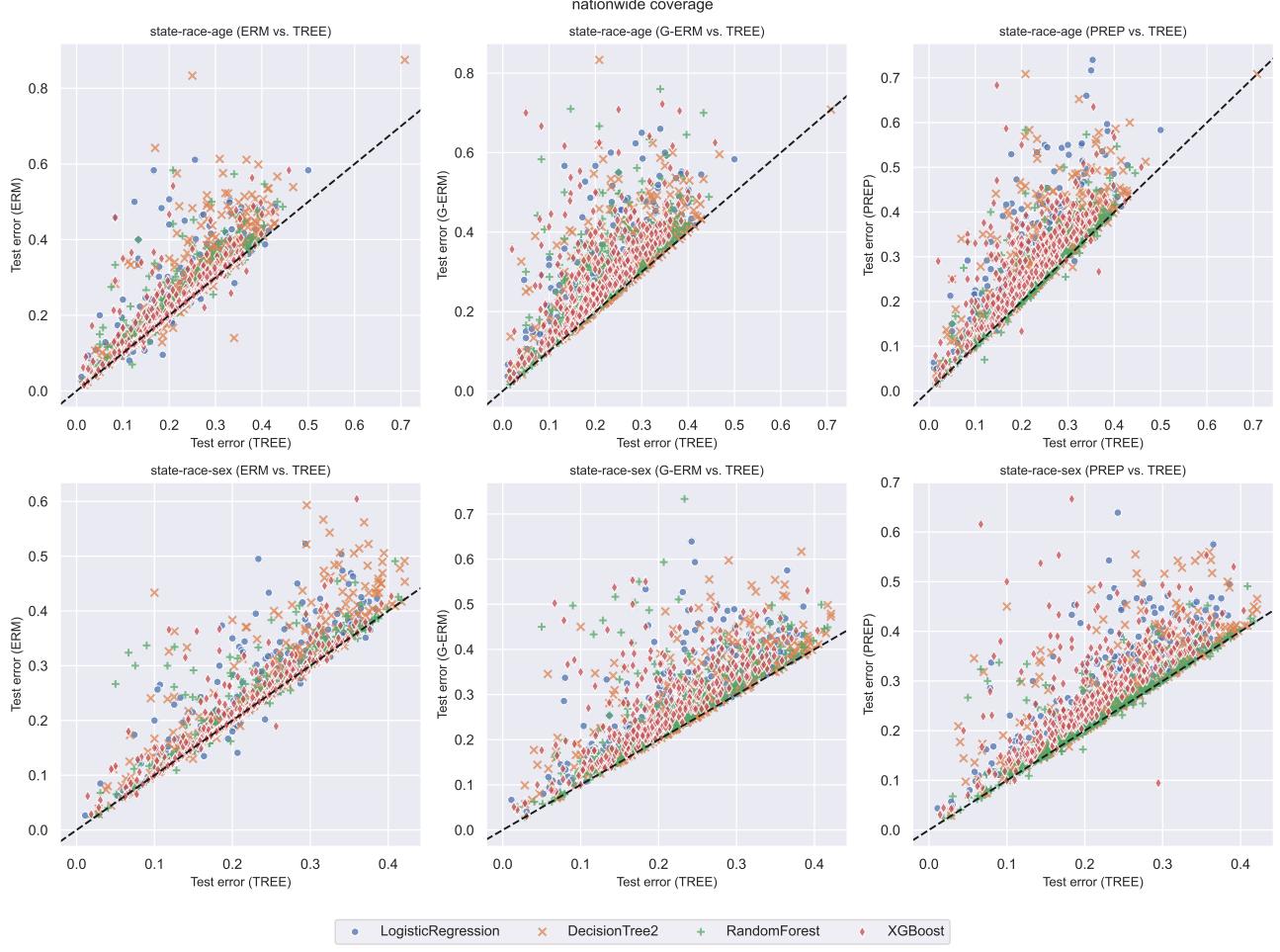
Figure 25. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups on nationwide dataset for the Income task. Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: decision trees with `max_depth` 2, decision trees with `max_depth` 4, decision trees with `max_depth` 8, and random forest.
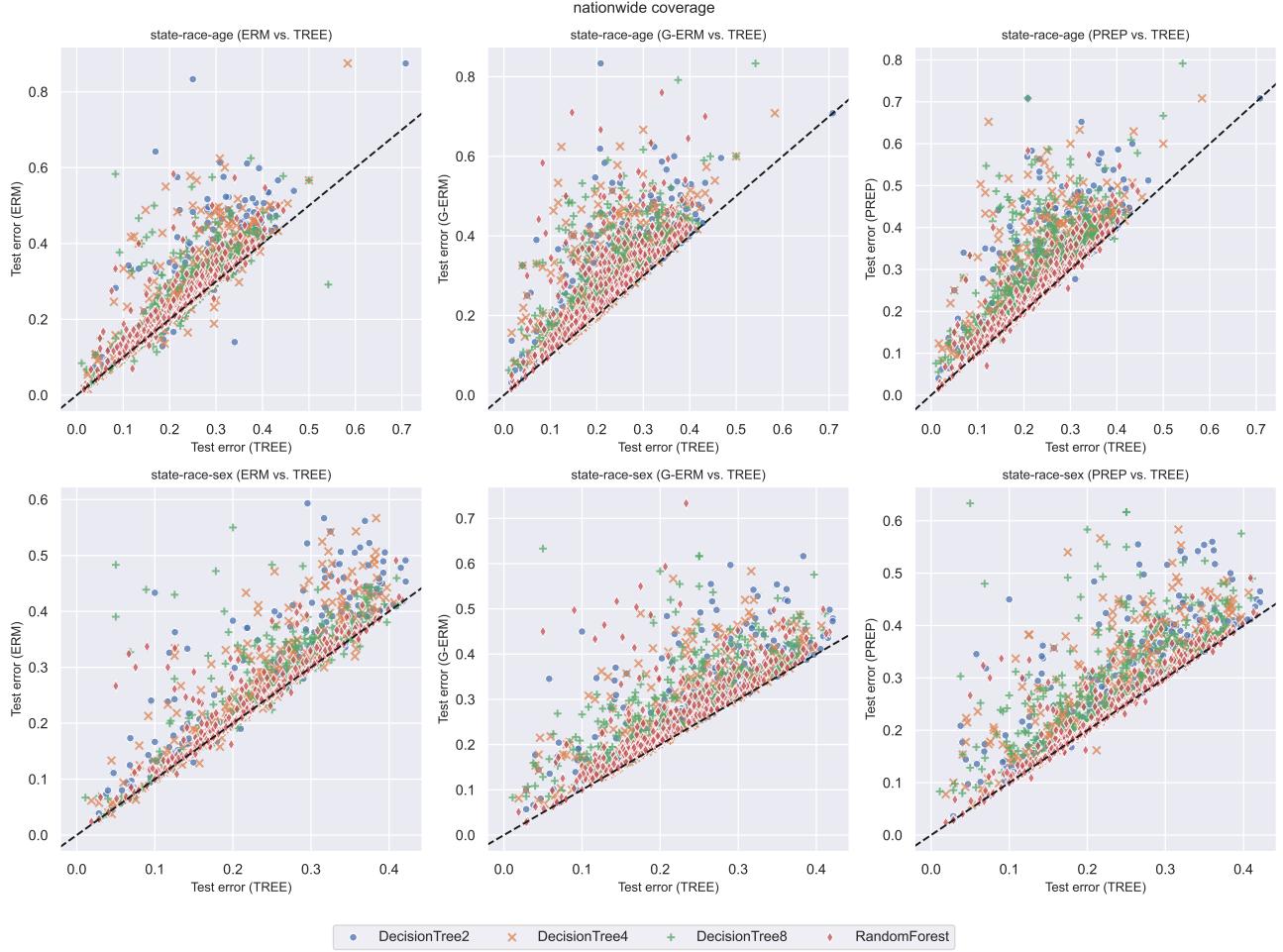
Figure 26. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups on nationwide dataset for the Employment task. Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: logistic regression, decision trees with `max_depth` 2, random forest, and XGBoost.
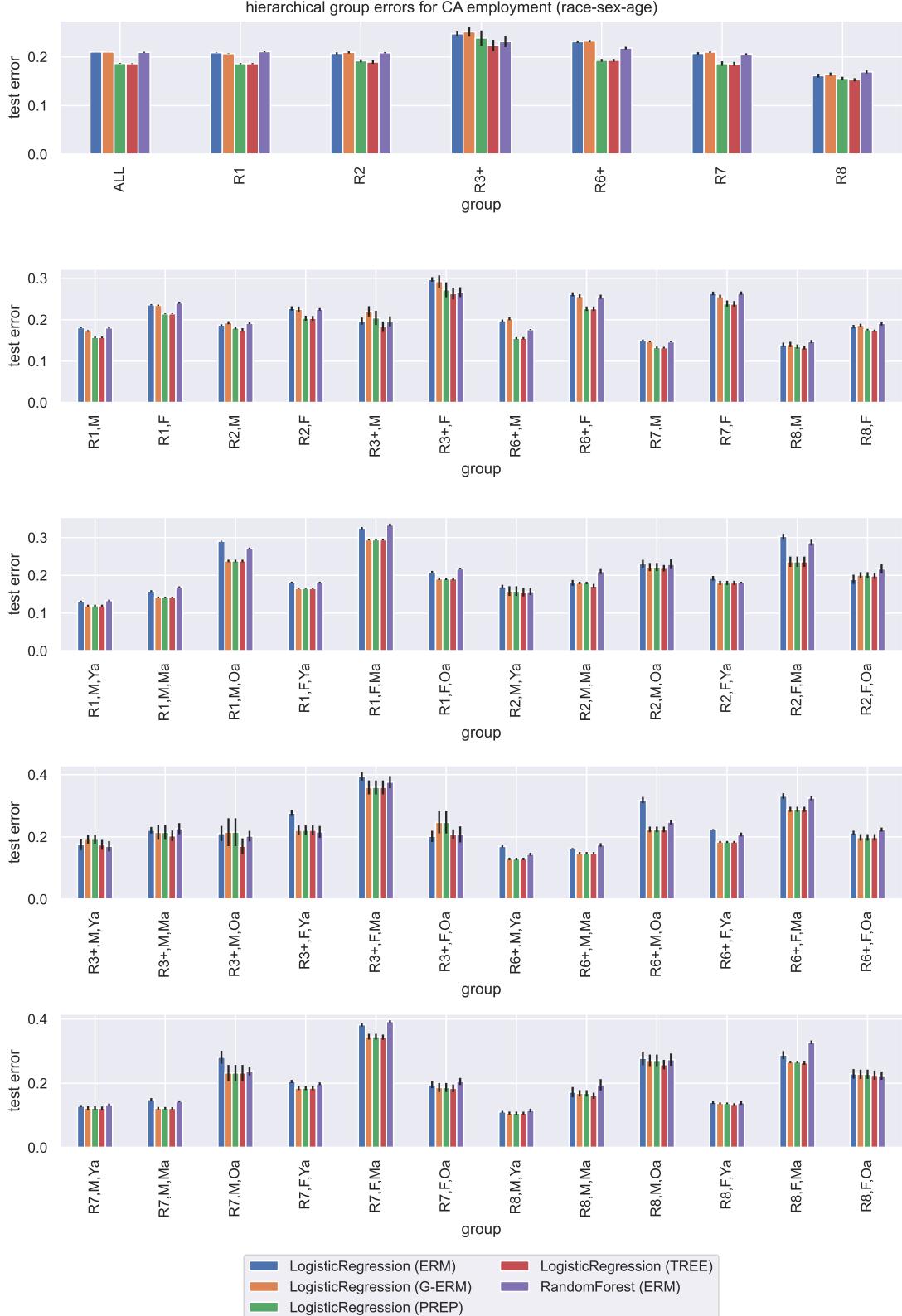
Figure 27. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups on nationwide dataset for the Employment task. Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: decision trees with `max_depth` 2, decision trees with `max_depth` 4, decision trees with `max_depth` 8, and random forest.

Figure 28. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups on nationwide dataset for the Coverage task. Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: logistic regression, decision trees with `max_depth` 2, random forest, and XGBoost.

Figure 29. Test error of `MGL-Tree` vs. ERM, Group ERM, and `Prepend` on `race-sex-age` and `race-sex-edu` groups on nationwide dataset for the Coverage task. Each point corresponds to a group; points above the $y = x$ line show that `MGL-Tree` generalizes better than the competitor method on that particular group. Benchmark hypothesis classes considered: decision trees with `max_depth` 2, decision trees with `max_depth` 4, decision trees with `max_depth` 8, and random forest.

Figure 5. **Test accuracy of $\mathcal{H} =$ Logistic Regression for `race-sex-age` groups (CA Employment).** Test errors across all $|\mathcal{G}| = 54$ hierarchically structured groups from `race-sex-age`. See Appendix F for more information on the specific categories for each group.