Group conditional validity via multi-group learning

Samuel Deng, Navid Ardeshir, and Daniel Hsu

Columbia University

January 30, 2023

Abstract

We consider the problem of distribution-free conformal prediction and the criterion of group conditional validity. This criterion is motivated by many practical scenarios including hidden stratification and group fairness. Existing methods achieve such guarantees under either restrictive grouping structure or distributional assumptions, or they are overly-conservative under heteroskedastic noise. We propose a simple reduction to the problem of achieving validity guarantees for individual populations by leveraging algorithms for a problem called multi-group learning. This allows us to port theoretical guarantees from multi-group learning to obtain obtain sample complexity guarantees for conformal prediction. We also provide a new algorithm for multi-group learning for groups with hierarchical structure. Using this algorithm in our reduction leads to improved sample complexity guarantees with a simpler predictor structure.

1 Introduction

The focus of this paper is sample-efficient algorithms for *distribution-free conformal prediction*. Consider a dataset $\mathcal{D}$ of random examples drawn i.i.d. from an unknown probability distribution over the domain $\mathcal{X} \times \mathcal{Y}$. The goal of conformal prediction is to use $\mathcal{D}$ to construct a *valid* set prediction function $C : \mathcal{X} \to 2^{\mathcal{Y}}$; by this, we mean that given the feature vector X of a new (independent) test example (X, Y) drawn from the same distribution, the predicted confidence region $C(X) \subseteq \mathcal{Y}$ should typically contain the corresponding label Y. We adopt the notion of (training conditional) validity introduced by Vovk [2012], Bian & Barber [2022][1]: the set predictor C is *valid at the level* $1 - \alpha$ if, with high probability over the draw of $\mathcal{D}$,

$$\Pr[Y \in C(X) \mid \mathcal{D}] \geq 1 - \alpha - o(1)$$

as the number of data becomes large. The coverage rate displayed in the left-hand side above is an average over possible test examples (X, Y).

In many cases, however, this notion of validity may be insufficient from the perspective of a particular test example (say, corresponding to a specific individual). This is because the average incorporates all possible test examples, many of which may have little resemblance to the particular

E-mail: samdeng@cs.columbia.edu, na2844@columbia.edu, djhsu@cs.columbia.edu

[1] Contrast this with *marginal validity*, which averages over both $\mathcal{D}$ and the test example. We drop the descriptor "training conditional" in the sequel for brevity, as we always give guarantees that hold with high probability over $\mathcal{D}$.

test example. The averaging therefore obscures the underlying structure of "coverage failures", as the confidence region might fail to contain the label for some examples more often than others. Similar issues were raised in practice for classification problems under the guise of hidden stratification Oakden-Rayner et al. [2020] and group fairness [Hardt et al., 2016, Agarwal et al., 2018, Donini et al., 2018, Blum & Lykouris, 2020].

The stricter requirement of *group conditional validity* is a natural way to address the aforementioned issues Vovk [2012], Foygel Barber et al. [2021]. Suppose $\mathcal{G} \subset 2^{\mathcal{X}}$ is a collection of (possibly overlapping) groups of feature vectors. For instance the groups could be based on various demographic characteristics, such as age, race, and gender. Or the groups could be feature-spatial regions (e.g., in $\mathbb{R}^d$) determined by a process such as dyadic partitioning. *Group conditional validity at level* $1-\alpha$ demands that, for each such group $g \in \mathcal{G}$, the conditional coverage rate for group g, defined by

$$\Pr[Y \in C(X) \mid X \in g, \mathcal{D}],$$

be at least $1 - \alpha - o(1)$. Note that *object conditional validity*—the special case where $\mathcal{G} = 2^{\mathcal{X}}$ is the maximally rich family of groups—is known to be impossible to non-trivially ensure in a distribution-free manner [Vovk, 2012, Lei & Wasserman, 2014]. Therefore, the main focus of this paper is group-conditional validity for groups $\mathcal{G}$ that are restricted in richness. For concreteness, we simply assume that $\mathcal{G}$ is finite but possibly very large.

Prior methods for group-conditional validity were initially developed for non-overlapping groups [Vovk, 2012] and then later for overlapping groups [Foygel Barber et al., 2021]. The method of Vovk uses (non-)conformity scores with the "Split Conformal" approach of Papadopoulos et al. [2002] on each group separately. Under the assumption of disjoint groups, one uses the predictive set function for the (unique) group containing X to construct the confidence region for X. To handle overlapping groups (in the context of regression), the method of Foygel Barber et al. considers the predictive intervals for all groups containing X, and then (essentially) uses the widest such interval. While this ensures validity, it may be overly conservative in problems with heteroskedastic noise [Jung et al., 2022]. Other methods address heteroskedasticity by aiming for stronger guarantees such as "multicalibration" and "multivalid coverage" [Jung et al., 2022], but these guarantees are achieved only under additional smoothness conditions on the distribution of conformity scores.

1.1 Contributions

Our goal is to develop algorithms that achieve distribution-free group-conditional validity. To this end, we propose and analyze a simple reduction to the simpler problem of achieving distribution-free validity on individual populations. Our reduction is based on *multi-group (agnostic PAC) learning* [Rothblum & Yona, 2021], a generalization of traditional agnostic PAC learning. In multi-group learning with respect to a family of groups $\mathcal{G}$ and a reference class of predictors $\mathcal{H}$, the goal is to produce a single predictor that, for every group $g \in \mathcal{G}$, has small excess *conditional risk* on a new random example (X, Y) given that $X \in g$ (see Section 3.1 for the formal definition). Our reduction, given in Section 3, can be instantiated with any learning algorithm designed for multi-group learning, and our analysis translates multi-group learning guarantees to guarantees on group-conditional validity. Instantiating this reduction with existing multi-group learning algorithms of Tosh & Hsu [2022] and standard conformal prediction methods [see, e.g., Vovk, 2012] yields new methods for building group-conditionally valid set predictors that apply more generally or are more

noise-adaptive than those from previous works.

A second contribution of this paper is a new multi-group learning algorithm for the important special case of groups that are hierarchically structured. The algorithm, presented in Section 4, produces simpler predictors (decision trees) with better statistical guarantees compared to that of some previous algorithms in the context of multi-group learning. We also explain how this procedure can have qualitative advantages over a black-box application of (the analysis of) our reduction.

1.2 Related work

Our work connects to the literature on conformal prediction and multi-group learning.

Conformal prediction. The distribution-free and model-agnostic guarantees offered by conformal prediction methods enable wide applicability in areas as varied as clinical research Lu et al. [2022], fairness in sentencing and recommendation Romano et al. [2020], Kuchibhotla & Berk [2021], and robotics Lindemann et al. [2022], Dixit et al. [2022]. The goal of general group-conditional validity (beyond the case of non-overlapping groups [Vovk, 2012]) was introduced and studied by Foygel Barber et al. [2021] under the name of "restricted conditional coverage". They show the sufficiency and necessity of bounded VC dimension on $\mathcal{G}$. As discussed above, the method proposed by Foygel Barber et al. may be overly conservative in situations with heteroskedastic noise, leading to confidence regions that are too large and uninformative. Jung et al. [2022] presents a method that is more adaptive to the group membership of new test examples, but it differs from our proposed methods in important ways. The method of Jung et al. aims for a guarantee called "multivalid coverage", which is qualitatively stronger than group-conditional validity. Consequently, their method relies on a Lipschitz condition on the distribution of conformity scores, which is not needed by our methods for achieving group-conditional validity. Their method's rate of convergence to the prescribed validity level is also asymptotically slower than that achievable using our reduction ($n^{-1/4}$ vs. $n^{-1/2}$).

Multi-group learning. The framework of multi-group agnostic PAC learning was formalized by Rothblum & Yona [2021], and their proposed methods were built atop the outcome indistinguishability framework of Dwork et al. [2021]. Multi-group agnostic PAC learning has an online counterpart which was studied by Blum & Lykouris [2020] in the context of group fairness. One of the algorithms of Tosh & Hsu [2022] gives the best convergence rates for multi-group learning, and hence via our reduction lead to our strongest guarantees for group-conditional validity. The algorithm we propose in Section 4 can be viewed as adaptation of the "Prepend" algorithm of Tosh & Hsu (and also of a related algorithm of Globus-Harris et al. [2022]). Our analysis of the algorithm in the special case of hierarchically structured groups is very different from that of "Prepend" and leads to an improved rate of convergence.

2 Preliminaries

2.1 Data and groups

Let $\mathcal{X}$ be the input space (e.g., of feature vectors describing instances), and let $\mathcal{Y}$ be the output space (e.g., of labels to predict). Let $\mathcal{G} \subseteq 2^{\mathcal{X}}$ be a (finite) family of groups $g \subseteq \mathcal{X}$, each of which is a subset of the input space. We overload notation and also use $g(x) = \mathbb{1}\{x \in g\}$ to denote the

indicator function for g. For a dataset $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$ and group g, let $\mathcal{D}_g = \{(x, y) \in \mathcal{D} : x \in g\}$ denote the set of examples in $\mathcal{D}$ whose feature vectors are in g. Similarly, for a probability distribution $\mathbb{P}$ over $\mathcal{X} \times \mathcal{Y}$, we let $\mathbb{P}_g$ denote the conditional distribution of $(X, Y) \sim \mathbb{P}$ given $X \in g$.

2.2 Notions of validity

Let $C \colon \mathcal{X} \to 2^{\mathcal{Y}}$ be a (possibly randomized) set predictor, constructed using a (random) dataset $\mathcal{D}$. For $\alpha, \delta \in (0, 1)$ and any probability distribution $\mathbb{P}$ over $\mathcal{X} \times \mathcal{Y}$, we say C is (α, δ)-*valid for* $\mathbb{P}$ if, with probability at least $1 - \delta$ over the random draw of $\mathcal{D}$,

$$\Pr[Y \in C(X) \mid \mathcal{D}] \geq 1 - \alpha, \tag{1}$$

where $(X, Y) \sim \mathbb{P}$ be a test example drawn from $\mathbb{P}$ (independent of $\mathcal{D}$). For any sequence $(\alpha_g : g \in \mathcal{G}) \in (0, 1)^{\mathcal{G}}$ and $\delta \in (0, 1)$, we say C is $((\alpha_g : g \in \mathcal{G}), \delta)$-*group conditionally valid for* $(\mathbb{P}, \mathcal{G})$ if, with probability at least $1 - \delta$ over the random draw of $\mathcal{D}$,

$$\Pr[Y \in C(X) \mid X \in g, \mathcal{D}] \geq 1 - \alpha_g \text{ for all } g \in \mathcal{G}. \tag{2}$$

We allow the level of validity $1 - \alpha_g$ to depend on the group $g \in \mathcal{G}$, e.g., through the probability mass of the group $\mathbb{P}(g)$, as the more examples one expects to have in $\mathcal{D}_g$, the greater the level of confidence one can expect to achieve.

2.3 Background on conformal prediction

Conformal prediction refers to techniques for constructing valid set predictors in a distribution-free manner (i.e., without assumptions on the unknown distribution $\mathbb{P}$), assuming only the availability of i.i.d. data from $\mathbb{P}$ [Vovk et al., 2005, Vovk, 2012]. The i.i.d. assumption can be relaxed to exchangeability, but for sake of brevity of the proofs, we stick to the i.i.d. setting.

One of the simplest approaches to conformal prediction that ensures validity (as defined in Section 2.2) is *Inductive Conformal Prediction (ICP; also called Split Conformal)* [Vovk, 2012], although several other holdout methods also enjoy this property [Bian & Barber, 2022]. We describe ICP below, following the presentation of Lei & Wasserman [2014], and also show a baseline approach to group-conditional validity with ICP.

Description of ICP. The ICP method uses a partition of observed data $\mathcal{D}$ into two datasets, $\mathcal{D}^{(\mathrm{tr})}$ and $\mathcal{D}^{(\mathrm{cal})}$. The training data $\mathcal{D}^{(\mathrm{tr})}$ is used to train a *predictor* $\mu : \mathcal{X} \to \mathcal{Z}$; the calibration data is used to build the set predictor $C_\alpha^{\mathrm{ICP}} \colon \mathcal{X} \to 2^{\mathcal{Y}}$ on top of μ for a given level $1 - \alpha$. (Typically $\mathcal{Z} = \mathcal{Y}$, but this is not strictly necessary.) ICP relies on a *non-conformity score* function $s : \mathcal{Y} \times \mathcal{Z} \to \mathbb{R}$ that compares predictor's outputs with true labels. A common example (in regression problems, where $\mathcal{Z} = \mathcal{Y} = \mathbb{R}$) is the residuals error of the prediction model $s(y, z) = |y - z|$.

The confidence regions produced by C_α^{ICP} are obtained by inverting tests of hypotheses of the form "$Y = y$", where $y \in \mathcal{Y}$ is a possible label of a given feature vector $x \in \mathcal{X}$. Forming p-values for such hypothesis tests is obtained using a non-parametric rank test on non-conformity scores; this is what enables the construction of distribution-free valid confidence regions. In detail, let $\hat{\mathbb{P}}^y$ be the empirical measure over $\mathcal{D}^{(\mathrm{cal})} \cup \{(x, y)\}$, which combines the calibration dataset $\mathcal{D}^{(\mathrm{cal})}$ with the provisioned test example (x, y). By forming scores $S(\tilde{Y}, \mu(\tilde{X}))$ where $(\tilde{X}, \tilde{Y}) \sim \hat{\mathbb{P}}^y$ and measuring the rank of the score corresponding to $S(y, \mu(x))$, we can construct a valid p-value

$$\hat{\mathbb{P}}^y[S(\tilde{Y}, \mu(\tilde{X})) \leq S(y, \mu(x))].$$

To achieve validity at level $1-\alpha$, the ICP set predictor outputs the set of all $y \in \mathcal{Y}$ for which the test is not rejected:

$$C_\alpha^{\mathrm{ICP}}(x; \mathcal{D}^{(\mathrm{cal})}, \mu) := \left\{ y \in \mathcal{Y} \,\middle|\, \hat{\mathbb{P}}^y[S(\tilde{Y}, \mu(\tilde{X})) \leq S(y, \mu(x))] > \alpha \right\}.$$

In the sequel, we drop the explicit dependence on $\mathcal{D}^{(\mathrm{cal})}$ and μ when it is clear from context.

Proposition 2.1 (Proposition 2a of Vovk, 2012)**.** *Given any $\alpha, \delta \in (0,1)$ and a dataset $\mathcal{D} = \mathcal{D}^{(\mathrm{tr})} \cup \mathcal{D}^{(\mathrm{cal})}$, where $\mathcal{D}^{(\mathrm{cal})}$ is drawn i.i.d. from $\mathbb{P}$ and independent of $\mathcal{D}^{(\mathrm{tr})}$, the set predictor C_α^{ICP} constructed by the ICP procedure described above is (α', δ)-valid for $\mathbb{P}$ with*

$$\alpha' = \alpha + \sqrt{\frac{\log(1/\delta)}{2|\mathcal{D}^{(\mathrm{cal})}|}}.$$

Of course, ICP can be applied to achieve validity for $\mathbb{P}_g$ for any individual group $g \in \mathcal{G}$, simply by using $\mathcal{D}_g^{(\mathrm{cal})}$ instead of $\mathcal{D}^{(\mathrm{cal})}$; this easily leads to a variant of Proposition 2.1 with $\mathbb{P}$ replaced by $\mathbb{P}_g$ and $|\mathcal{D}^{(\mathrm{cal})}|$ replaced by $|\mathcal{D}_g^{(\mathrm{cal})}|$.[2]

Baseline approach to group-conditional validity. Recall that group-conditional validity requires a single set predictor C that ensures a prescribed level of validity for all groups simultaneously. In the case where the groups in $\mathcal{G}$ are disjoint (and, for simplicity, assumed to satisfy $\bigcup_{g \in \mathcal{G}} = \mathcal{X}$), then this is easy to achieve by constructing a separate set predictor C^g for each $g \in \mathcal{G}$ (say, using ICP), and then defining $C(x) = C^g(x)$ where g is the unique group $g \in \mathcal{G}$ that contains x [Vovk, 2012]. However, if groups in $\mathcal{G}$ may overlap, then this approach does not work, as it is not clear what confidence region to return for any x that belongs to more than one group.

A baseline solution to this problem is to replace $\mathcal{G}$ with a family $\mathcal{G}^*$ of non-overlapping groups that refines $\mathcal{G}$, and then to use the method described above with $\mathcal{G}^*$. However, the level of validity $1-\alpha_g$ achievable for a group $g \in \mathcal{G}$ will suffer with this approach if g is shattered into many smaller groups $g = g_1 \cup \cdots \cup g_k$ in this new family, where the $g_i \in \mathcal{G}^*$ may only have a small fraction of the $\mathbb{P}$-mass of g. We show in the next section that such a degradation can be avoided by working only with individual set predictors C^g for each $g \in \mathcal{G}$.

3 Reduction based on multi-group learning

In this section, we describe and analyze a reduction from the problem of learning group-conditionally valid set predictors to the problem of valid set predictors for single populations. The reduction is based on algorithms for a problem called multi-group learning. We instantiate this reduction with a particular multi-group algorithm and the ICP method from Section 2.3 to give an end-to-end procedure for achieving group-conditional validity.

[2]There is one technical difference, which is that $|\mathcal{D}_g^{(\mathrm{cal})}|$ is technically a (binomial) random variable, so the level of validity $1-\alpha'$ would also technically be random. However, the concentration properties of $|\mathcal{D}_g^{(\mathrm{cal})}|$ are well-understood, and it is possible to write an achievable level of validity purely in terms of $|\mathcal{D}^{(\mathrm{cal})}|$ and $\mathbb{P}(g)$ (and (α, δ)). We forgo this in this paper for sake of simplicity.

3.1 Background on multi-group learning

Multi-group learning, formalized by Rothblum & Yona [2021], is a generalization of the traditional agnostic PAC learning setup. In agnostic PAC learning [Kearns et al., 1992, Haussler, 1992], the goal is to produce a predictor that, with high probability, achieves small excess risk (relative to a benchmark class of predictors). In multi-group learning, the goal is to achieve small excess conditional risk, simultaneously over all groups $g \in \mathcal{G}$.

For $\mathbb{P}$ be a probability distribution over $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{H}$ be a *benchmark class* of predictors $h : \mathcal{X} \to \mathcal{A}$, which serves as a family of benchmark predictors; also let $\ell : \mathcal{A} \times \mathcal{Y} \to [0, 1]$ be a bounded loss function for measuring the quality of predictions. The *conditional ℓ-risk* of a (possibly randomized) predictor $f \colon \mathcal{X} \to \mathcal{Y}$ for group $g \subseteq \mathcal{X}$ is

$$L(f \mid g) := \mathbb{E}\left[\ell(f(X), Y) \mid X \in g\right],$$

where the conditional expectation is over $(X, Y) \sim \mathbb{P}_g$ (and any possible internal randomization of f). The excess conditional ℓ-risk of f for group g (relative to $\mathcal{H}$) is $L(f \mid g) - \inf_{h \in \mathcal{H}} L(h \mid g)$. A *multi-group learning (MGL)* algorithm for $(\mathcal{G}, \mathcal{H})$ is given a dataset $\mathcal{D} \subseteq \mathcal{X} \times \mathcal{Y}$ of random examples drawn i.i.d. from $\mathbb{P}$, and returns a predictor $f \colon \mathcal{X} \to \mathcal{A}$ (which may be randomized, and also need not be in $\mathcal{H}$). The goal of such an algorithm is to ensure f has small excess conditional ℓ-risk for all groups $g \in \mathcal{G}$ simultaneously.

Definition 3.1 (MGL property)**.** A multi-group learning algorithm for $(\mathcal{G}, \mathcal{H})$ satisfies the *MGL property* with excess conditional ℓ-risk bounds $\epsilon_g(\cdot, \cdot)$ for $g \in \mathcal{G}$ if it returns a (possibly randomized) predictor $f \colon \mathcal{X} \to \mathcal{A}$ such that, with probability at least $1 - \delta$ over the random draw of $\mathcal{D}$,

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

Above, $\epsilon_g(\mathcal{D}, \delta)$ bounds the excess conditional ℓ-risk of f for group g, and it is typically decreasing in $|\mathcal{D}_g|$ and δ. (It may also depend on $\mathcal{H}$ and $\mathcal{G}$.)

Tosh & Hsu [2022] presents two MGL algorithms satisfying the MGL property with different excess conditional ℓ-risk bounds. Here we quote the guarantee for one of these algorithms, which is based on an online-to-batch conversion of a "sleeping experts" online learning algorithm of Blum & Mansour [2007].

Proposition 3.2 (Theorem 10 of Tosh & Hsu, 2022)**.** *For any bounded loss function ℓ, there is an MGL algorithm for $(\mathcal{G}, \mathcal{H})$ that satisfies the MGL property with excess conditional ℓ-risk bounds*

$$\epsilon_g(\mathcal{D}, \delta) = O\left(\sqrt{\frac{\log(|\mathcal{H}||\mathcal{G}|/\delta)}{|\mathcal{D}_g|}}\right) \quad \textit{for all } g \in \mathcal{G}.$$

3.2 The reduction

We now give our reduction based on multi-group learning.

Let $\mathcal{H} = \{C^g : g \in \mathcal{G}\}$ be a benchmark class of set predictors, one per group in $\mathcal{G}$. (We henceforth use $\mathcal{A} = 2^{\mathcal{Y}}$.) Our reduction is simply to run an MGL algorithm for $(\mathcal{G}, \mathcal{H})$, where the loss function is the 0/1 "miscoverage" loss function $\ell_{0/1} \colon 2^{\mathcal{Y}} \times \mathcal{Y} \to [0, 1]$ defined by

$$\ell_{0/1}(c, y) := \mathbb{1}\{y \notin c\}. \tag{3}$$

The dataset provided to the MGL algorithm is a separate dataset $\mathcal{D}^{(\mathrm{mg})}$, drawn i.i.d. from $\mathbb{P}$ and independent of data used to construct the set predictors in $\mathcal{H}$. The MGL algorithm returns a set predictor $C\colon \mathcal{X} \to 2^{\mathcal{Y}}$. We show below that C satisfies group-conditional validity for $\mathbb{P}$ as long as the set predictors in $\mathcal{H}$ are valid for their respective conditional distributions $\mathbb{P}_g$'s.

Theorem 3.3. *Suppose the set predictors in $\mathcal{H} = \{C^g : \mathcal{X} \to 2^{\mathcal{Y}} : g \in \mathcal{G}\}$ are constructed using a dataset $\mathcal{D}$ and each C^g is $(\alpha_g, \delta/(2|\mathcal{G}|))$-valid for $\mathbb{P}_g$. Let $C\colon \mathcal{X} \to 2^{\mathcal{Y}}$ be the set predictor returned by a MGL algorithm A for $(\mathcal{G}, \mathcal{H})$ on a dataset $\mathcal{D}^{(\mathrm{mg})}$ of i.i.d. random examples from $\mathbb{P}$, independent of $\mathcal{D}$. If A satisfies the MGL property with excess conditional $\ell_{0/1}$-risk bounds $\epsilon_g(\cdot,\cdot)$ for $g \in \mathcal{G}$, then C is $((\alpha'_g : g \in \mathcal{G}), \delta)$-valid for $(\mathbb{P}, \mathcal{G})$ with*

$$\alpha'_g = \alpha_g + \epsilon_g(\mathcal{D}^{(\mathrm{mg})}, \delta/2) \quad \text{for all } g \in \mathcal{G}.$$

Theorem 3.3 is a direct consequence of the definitions given above, the particular choice of the loss function, and union bounds to ensure that the "failure probability" in the definition of group-conditional validity is at most δ.

3.3 End-to-end algorithm for group-conditional validity

We demonstrate the utility of our reduction by using it to give an end-to-end procedure that produces group-conditionally valid set predictors.

1. Let $\mathcal{D}$ be a dataset of random examples drawn i.i.d. from $\mathbb{P}$, partitioned into three sets, $\mathcal{D}^{(\mathrm{tr})}$, $\mathcal{D}^{(\mathrm{cal})}$, and $\mathcal{D}^{(\mathrm{mg})}$.

2. For each group $g \in \mathcal{G}$, use $\mathcal{D}^{(\mathrm{tr})}$ to train a predictor $\mu_g\colon \mathcal{X} \to \mathcal{Z}$, and then use $\mathcal{D}^{(\mathrm{cal})}$ and μ_g to construct a set predictor $C_g\colon \mathcal{X} \to 2^{\mathcal{Y}}$ using ICP with nominal level $1-\alpha$:

$$C^g = C_\alpha^{\mathrm{ICP}}(\,\cdot\,; \mathcal{D}_g^{(\mathrm{cal})}, \mu_g), \tag{4}$$

   and let $\hat{\mathcal{H}} := \{C^g : g \in \mathcal{G}\}$.[3]

3. Run an MGL algorithm A from Proposition 3.2 for $(\mathcal{G}, \hat{\mathcal{H}})$ and the loss function $\ell_{0/1}$ using the dataset $\mathcal{D}^{(\mathrm{mg})}$ to obtain the final set predictor $C\colon \mathcal{X} \to 2^{\mathcal{Y}}$.

We have the following corollary of Theorem 3.3 (by way of Proposition 3.2).

Corollary 3.4. *Let $C\colon \mathcal{X} \to 2^{\mathcal{Y}}$ be the set predictor constructed by the procedure described above. With probability at least $1-\delta$ over the random draw of $\mathcal{D}$, C is $((\alpha'_g : g \in \mathcal{G}), \delta)$-valid for $(\mathbb{P}, \mathcal{G})$ with*

$$\alpha'_g = \alpha + c\sqrt{\left(\frac{\log(|\mathcal{G}|/\delta)}{|\mathcal{D}_g^{(\mathrm{cal})}|} + \frac{\log(|\mathcal{G}|/\delta)}{|\mathcal{D}_g^{(\mathrm{mg})}|}\right)},$$

where $c > 0$ is a universal constant.

[3]Note that $\hat{\mathcal{H}}$ is a random benchmark class that is $\sigma(\mathcal{D}^{(\mathrm{tr})}, \mathcal{D}^{(\mathrm{cal})})$-measurable. The dependence of $\hat{\mathcal{H}}$ on the training data is only through the predictor functions $\{\mu_g\}_{g\in\mathcal{G}}$.

We note that, for each $x \in \mathcal{X}$, the predicted confidence region $C(x)$ is randomly chosen from among $\{C^g(x) : g \in \mathcal{G}, x \in g\}$, with each region having a positive chance of being picked. Below, we give an example in the context of regression (where $\mathcal{Y} = \mathbb{R}$) to illustrate qualitatively how our procedure can be more adaptive that of Foygel Barber et al. [2021]. Recall that their method (essentially) picks the "widest" confidence region corresponding to groups that contains the test feature vector, and this always leads to group-conditionally valid confidence intervals. This turns out to be asymptotically optimal in some scenarios with homogeneous noise, but it may be suboptimal when heteroskedastic noise is present.

Example. We construct a data distribution $\mathbb{P}$ in which the label Y has different variance under the different group-conditional distributions $\mathbb{P}_g$. In particular, for any $x \in \mathcal{X}$, we let the conditional distribution of Y given $X = x$ in $\mathbb{P}$ be zero-mean normal

$$Y \mid X = x \ \sim \ \mathcal{N}(0, \sigma^2(x))$$

with variance depending on x as follows:

$$\sigma^2(x) = \begin{cases} v_1 & \text{if } x \in g_1 \setminus g_2 \\ v_2 & \text{if } x \in g_2 \setminus g_1 \\ v_{1,2} & \text{if } x \in g_1 \cap g_2 \end{cases}$$

for some positive values $v_1, v_2, v_{1,2}$. (Here, $\mathcal{G} = \{g_1, g_2\}$.) We let the marginal distribution of X be one in which all possible combinations of group memberships are equally likely, so $(g_1(X), g_2(X))$ is a uniformly random in $\{0,1\}^2$.

Suppose that set predictors C^{g_1} and C^{g_2} are formed using ICP with the "absolute residual" nonconformity score function $S(y,z) = |y - z|$ and also the same (constant) global predictor $\mu_g(x) \equiv 0$ for both groups $g \in \mathcal{G}$. For the distribution above, these set predictors will be qualitatively different, as C^{g_i} outputs a confidence interval of (square) length roughly proportional to

$$\mathrm{var}(Y \mid X \in g_i) = \frac{1}{2} v_i + \frac{1}{2} v_{1,2}.$$

Suppose $v_{1,2} \ll v_1 \ll v_2$, so the confidence interval for group g_1 is very short compared to that of group g_2. Picking the widest confidence interval will, half of the time, lead to over-coverage for $X \in g_1$, as the long interval for group g_2 will be for all $X \in g_1 \cap g_2$. However, the set predictor produced using the procedure of Corollary 3.4 will use the shorter interval for g_1 at least a fraction of the time, even for $X \in g_1 \cap g_2$. This will leads to confidence intervals of smaller average length.

4 Multi-group learning with hierarchical group structure

One potential drawback of the algorithm from Corollary 3.4 is that the resulting set predictor is fairly complex and randomized. To address this drawback, we could instead apply our reduction using a different multi-group learning algorithm, such as the "Prepend" algorithm of Tosh & Hsu [2022], which outputs a deterministic decision list predictor. However, Prepend has the MGL property only with

$$\epsilon_g(\mathcal{D}, \delta) = O\left(\sqrt[3]{\frac{\log(|\mathcal{H}||\mathcal{G}|/\delta)}{\gamma |\mathcal{D}_g|}} \right),$$

where $\gamma := \min_{g \in \mathcal{G}} \mathbb{P}(g)$. This cube-root rate should be compared to the square-root rate in Proposition 3.2; also note the lack of an explicit dependence on γ in Proposition 3.2. Using Prepend implies a slower rate-of-convergence to the nominally prescribed validity level compared to that from Corollary 3.4.

In this section, we show that under the natural assumption that $\mathcal{G}$ is *hierarchically structured*, there is a different procedure Algorithm 1—which we call `MGL-Tree-Conformal`—that recovers the same rate-of-convergence as that from Corollary 3.4, but outputs a much simpler (and deterministic) *decision tree* predictor. Algorithm 1 is based on a new multi-group learning algorithm, `MGL-Tree` (Algorithm 2), which we describe in full generality in Appendix A.

4.1 Hierarchical group structure

A family of groups $\mathcal{G} \subseteq 2^{\mathcal{X}}$ is *hierarchically structured* (also called *laminar*) if, for every pair of distinct groups $g, g' \in \mathcal{G}$, exactly one of the following holds:

- $g \cap g' = \emptyset$ (g and g' are disjoint),
- $g \subset g'$ (g is contained in g'),
- $g' \subset g$ (g' is contained in g).

Put another way, the groups in $\mathcal{G}$ can be arranged as nodes in a rooted tree $\mathcal{T}_{\mathcal{G}}$ such that if g is a child of g' in $\mathcal{T}_{\mathcal{G}}$, then $g \subset g'$; and if g and g' are siblings in $\mathcal{T}_{\mathcal{G}}$, then $g \cap g' = \emptyset$. For simplicity, we assume that $\mathcal{X}$ itself is in $\mathcal{G}$. In this case, $\mathcal{X}$ is the root of $\mathcal{T}_{\mathcal{G}}$, and the leaves of any pruning of $\mathcal{T}_{\mathcal{G}}$ form a partitioning of some subset of $\mathcal{X}$.

Hierarchically structured groups naturally arise in many situations, such as when a base set $\mathcal{X}$ is recursively partitioned into categories and further subcategories. In a machine learning context, they arise when groups are obtained using agglomerative clustering of a dataset or a dyadic partitioning of a feature space.

The hierarchical structure (via $\mathcal{T}_{\mathcal{G}}$) suggests a very natural structure for a set predictor $C \colon \mathcal{X} \to 2^{\mathcal{Y}}$ as follows. Each node in $\mathcal{T}_{\mathcal{G}}$ is annotated with a predictor $\hat{C}^g \colon \mathcal{X} \to 2^{\mathcal{Y}}$, and to compute the output of the C on input x, simply find the "deepest" g in $\mathcal{T}_{\mathcal{G}}$ that contains x by following a path starting at the root $\mathcal{X}$ and moving from parent to child whenever the child contains x; the prediction $C(x)$ is then taken to be $\hat{C}^g(x)$. This is essentially a decision tree, with the one small wrinkle that an input x need not be routed all the way to a leaf of $\mathcal{T}_{\mathcal{G}}$; instead, it may stop at some intermediate tree node. Nevertheless, the structure of C is as natural as that of a (more standard) decision tree.

4.2 Group conditional validity from `MGL-Tree-Conformal`

Algorithm 1 ultimately outputs a decision tree set predictor, $C : \mathcal{X} \to 2^{\mathcal{Y}}$, of the form described above. We now sketch how that decision tree is constructed, using the notation from Section 3 and the tree $\mathcal{T}_{\mathcal{G}}$ above. Recall that each node of C is associated with a group $g \in \mathcal{G}$. As in Section 3, suppose we have a collection $\{C^g : g \in \mathcal{G}\}$ of set predictors C^g each based on dataset $\mathcal{D}$ of i.i.d. examples from $\mathbb{P}$, and C^g is $(\alpha_g, \delta/(2|\mathcal{G}|))$-valid for $\mathbb{P}_g$.

We just need to decide how to annotate each node g in C with a set predictor $\hat{C}^g \colon \mathcal{X} \to 2^{\mathcal{Y}}$. For now, we just describe how to do this with a separate, independent dataset $\mathcal{D}^{(\mathrm{mg})}$ of i.i.d. examples from $\mathbb{P}$. Initially, we set $\hat{C}^{\mathcal{X}} = C^{\mathcal{X}}$, the set predictor that is valid for $\mathbb{P} = \mathbb{P}_{\mathcal{X}}$. Then, we consider nodes g in $\mathcal{T}_{\mathcal{G}}$ in breadth-first order starting at the root $\mathcal{X}$ to determine the remaining $\hat{C}^g$. At node

g, we compute the difference between the *empirical group-conditional miscoverage rates* of $\hat{C}^{\mathrm{pa}(g)}$ and C^g, where $\mathrm{pa}(g)$ is the parent of g in $\mathcal{T}_\mathcal{G}$. To write this, first define

$$\Delta_g(x, y) := \mathbb{1}\{y \notin \hat{C}^{\mathrm{pa}(g)}(x)\} - \mathbb{1}\{y \notin C^g(x)\}$$

for an example $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Then, the difference in empirical group-conditional miscoverage rates on $\mathcal{D}^{(\mathrm{mg})}$ can be written as

$$\mathrm{Diff}_g := \frac{1}{|\mathcal{D}_g^{(\mathrm{mg})}|} \sum_{(X,Y) \in \mathcal{D}_g^{(\mathrm{mg})}} \Delta_g(X, Y). \tag{5}$$

We set $\hat{C}^g$ to be either C^g or $\hat{C}^{\mathrm{pa}(g)}$ depending on the value of Diff_g. Let

$$\varepsilon_g(\mathcal{D}^{(\mathrm{mg})}, \delta) := 18\sqrt{\frac{4\log(|\mathcal{G}|) + \log(8/\delta)}{|\mathcal{D}_g^{(\mathrm{mg})}|}}. \tag{6}$$

If the empirical group-conditional miscoverage rate of C^g is better than that of $\hat{C}^{\mathrm{pa}(g)}$ by a large-enough margin—specifically, if $\mathrm{Diff}_g \geq \varepsilon_g(\mathcal{D}^{(\mathrm{mg})}, \delta)$—then we set $\hat{C}^g := C^g$. If not, we set $\hat{C}^g := \hat{C}^{\mathrm{pa}(g)}$. This is described in full in Algorithm 1.

Algorithm 1 `MGL-Tree-Conformal`

Input: Family of hierarchically structured groups $\mathcal{G} \subseteq 2^\mathcal{X}$; collection of set predictors $\{C^g : g \in \mathcal{G}\}$ based on $\mathcal{D}$; dataset $\mathcal{D}^{(\mathrm{mg})}$ of i.i.d. random examples from $\mathbb{P}$, independent of $\mathcal{D}$; probability parameter $\delta \in (0, 1)$.

Output: Decision tree set predictor $C : \mathcal{X} \to 2^\mathcal{Y}$.

Set $\hat{C}^\mathcal{X} := C^\mathcal{X}$.
for each $g \in \mathcal{G} \setminus \{\mathcal{X}\}$ in breadth-first order **do**
  Compute Diff_g and $\varepsilon_g(\mathcal{D}^{(\mathrm{mg})}, \delta)$ (from (5) and (6))
  if $\mathrm{Diff}_g > \varepsilon_g(\mathcal{D}^{(\mathrm{mg})}, \delta)$ **then**
    Set $\hat{C}^g := C^g$.
  else
    Set $\hat{C}^g := \hat{C}^{\mathrm{pa}(g)}$.
  end if
end for
return Set predictor C as described in Section 4.1.

Theorem 4.1. *Suppose the collection of set predictors $\{C^g : g \in \mathcal{G}\}$ are constructed using a dataset $\mathcal{D}$ of i.i.d. random examples from $\mathbb{P}$ so that for each $g \in \mathcal{G}$, C^g is $(\alpha_g, \delta/(2|\mathcal{G}|))$-valid for $\mathbb{P}_g$. Let C be the decision tree set predictor returned by Algorithm 1 using a dataset $\mathcal{D}^{(\mathrm{mg})}$ of i.i.d. random examples from $\mathbb{P}$ independent of $\mathcal{D}$. Then C is $((\alpha'_g : g \in \mathcal{G}), \delta)$-group conditionally valid for $(\mathbb{P}, \mathcal{G})$ with*

$$\alpha'_g = \alpha_g + c\sqrt{\frac{\log(|\mathcal{G}|/\delta)}{|\mathcal{D}_g^{(\mathrm{mg})}|}}$$

where $c > 0$ is a universal constant.

If ICP is used construct the set predictors $\{C^g : g \in \mathcal{G}\}$ with Algorithm 1, then we obtain the following end-to-end group-conditional validity guarantee as a corollary of Theorem 4.1.

Corollary 4.2. *Suppose that the set predictors $\{C^g : g \in \mathcal{G}\}$ are generated by ICP with nominal level $1-\alpha$ using a dataset $\mathcal{D} = \mathcal{D}^{(\mathrm{tr})} \cup \mathcal{D}^{(\mathrm{cal})}$ of i.i.d. random examples from $\mathbb{P}$. Let C be the decision tree set predictor returned by Algorithm 1 using a dataset $\mathcal{D}^{(\mathrm{mg})}$ of i.i.d. random examples from $\mathbb{P}$ independent of $\mathcal{D}$. Then C is $((\alpha'_g : g \in \mathcal{G}), \delta)$-group conditionally valid for $(\mathbb{P}, \mathcal{G})$, with*

$$\alpha'_g = \alpha + c\sqrt{\frac{\log(|\mathcal{G}|/\delta)}{|\mathcal{D}_g^{(\mathrm{cal})}|} + \frac{\log(|\mathcal{G}|/\delta)}{|\mathcal{D}_g^{(\mathrm{mg})}|}},$$

where $c > 0$ is a universal constant.

Reducing the number of data splits. The overall procedure described above in Corollary 4.2 still uses three datasets: $\mathcal{D}^{(\mathrm{tr})}$ for training the group-specific predictors μ_g, $\mathcal{D}^{(\mathrm{cal})}$ for constructing the valid set predictors C^g via ICP, and $\mathcal{D}^{(\mathrm{mg})}$ for forming the final set predictor C. This is precisely the situation that we have by using the reduction from Section 3 with Algorithm 2 (in Appendix A), our general MGL algorithm for hierarchically structured groups.

However, it is possible, and potentially advantageous, to re-use $\mathcal{D}^{(\mathrm{tr})}$ in place of $\mathcal{D}^{(\mathrm{mg})}$. Specifically, we may use $\mathcal{D}^{(\mathrm{tr})}$ in Algorithm 1 in place of $\mathcal{D}^{(\mathrm{mg})}$ to estimate the miscoverage rates of C^g on the various group-conditional distributions $\mathbb{P}_{g'}$. If the family of predictors μ_g's underlying the set predictors C^g's is taken from a class of predictors that is not too rich, then these empirical miscoverage rates will converge uniformly to the actual miscoverage rates as $\mathcal{D}^{(\mathrm{tr})}$ becomes large. In this case, we obtain a similar result as in Corollary 4.2, with $\log(|\mathcal{G}|/\delta)/|\mathcal{D}_g^{(\mathrm{mg})}|$ replaced by $\log(|\mathcal{M}|/\delta)/|\mathcal{D}_g^{(\mathrm{tr})}|$, where $\mathcal{M}$ is the class of predictors from which the μ_g's are taken. The cardinality of $\mathcal{M}$ can be replaced by other capacity measures (e.g., covering numbers) that enable uniform convergence. This is a pessimistic estimate, since the criterion used for selecting the μ_g's is not necessarily strongly correlated with the miscoverage rates of the corresponding C^g's.

By avoiding the need for $\mathcal{D}^{(\mathrm{mg})}$, we avoid the need to split available data into three parts, and instead only require a split into two parts ($\mathcal{D}^{(\mathrm{tr})}$ and $\mathcal{D}^{(\mathrm{cal})}$, same as the standard ICP method). This may be practically important because it leaves more data available for training the predictors or for performing the conformalization in ICP.

5 Conclusion and future work

Our work provides new mechanisms for obtaining distribution-free group conditionally valid guarantees by reducing to the simpler problem of validity on individual populations of interest. Our reduction is based on new algorithms for the problem of multi-group learning, and this reduction-based approach is able to take advantage of preexisting standard conformal prediction methods. Moreover, we provide a new, simple algorithm for the case of hierarchically structured groups, with improved statistical guarantees compared to previous existing algorithms.

Below, we discuss a few directions for future work.

- Algorithm 1 takes advantage of the structure of $\mathcal{G}$ in a natural way, but oftentimes additional or different prior information about groups may be available. Prior knowledge about the similarity between groups (e.g., in terms of how similar their optimal predictors are, or how

similar their label variances are) may be desirable to incorporate in the construction of the overall set predictor. For instance, if a pair of groups g and g' are found to be similar in this way, perhaps it would be beneficial to augment $\mathcal{G}$ with $g \cup g'$.

- Group-conditional guarantees have been connected to robustness in the context of hidden stratification [Oakden-Rayner et al., 2020]: a group-conditional guarantee with a family $\mathcal{G}$ that is closed under complements implies robustness to attacks on individual groups of training data. Although there is a vast body of literature on distributionally-robust optimization [e.g., Duchi et al., 2021, Hu et al., 2018, Sagawa et al., 2020], it has yet to be connected to multi-group learning or group-conditional validity in conformal prediction.
- Our work also gives a "proof of concept" for the generality of the PAC-style statistical guarantees achievable in the multi-group learning framework. Conformal prediction is a particularly nice application of this (as group conditional validity is a desirable property for conformal predictors), but the techniques in our reductions may find other applications where group structure is important to consider.

References

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. A reductions approach to fair classification. In *International Conference on Machine Learning*, pp. 60–69, 2018.

Bian, M. and Barber, R. F. Training-conditional coverage for distribution-free predictive inference. *arXiv preprint arXiv:2205.03647*, 2022.

Blum, A. and Lykouris, T. Advancing subgroup fairness via sleeping experts. In *Innovations in Theoretical Computer Science Conference (ITCS)*, 2020.

Blum, A. and Mansour, Y. From external to internal regret. *Journal of Machine Learning Research*, 8(6), 2007.

Dixit, A., Lindemann, L., Wei, S., Cleaveland, M., Pappas, G. J., and Burdick, J. W. Adaptive conformal prediction for motion planning among dynamic agents. 2022. doi: 10.48550/ARXIV. 2212.00278. URL `https://arxiv.org/abs/2212.00278`.

Donini, M., Oneto, L., Ben-David, S., Shawe-Taylor, J., and Pontil, M. Empirical risk minimization under fairness constraints. In *Advances in Neural Information Processing Systems*, pp. 2796–2806, 2018.

Duchi, J. C., Glynn, P. W., and Namkoong, H. Statistics of robust optimization: A generalized empirical likelihood approach. *Mathematics of Operations Research*, 2021.

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 Co mputing*, pp. 1095–1108, 2021.

Foygel Barber, R., Candes, E. J., Ramdas, A., and Tibshirani, R. J. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021.

Globus-Harris, I., Kearns, M., and Roth, A. Beyond the frontier: Fairness without privacy loss. *arXiv preprint arXiv:2201.10408*, 2022.

Hardt, M., Price, E., and Srebro, N. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

Haussler, D. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and computation*, 100(1):78–150, 1992.

Hu, W., Niu, G., Sato, I., and Sugiyama, M. Does distributionally robust supervised learning give robust classifier s? In *International Conference on Machine Learning*, pp. 2029–2037. PMLR, 2018.

Jung, C., Noarov, G., Ramalingam, R., and Roth, A. Batch multivalid conformal prediction. 2022. URL `https://arxiv.org/abs/2209.15145`.

Kearns, M. J., Schapire, R. E., and Sellie, L. M. Toward efficient agnostic learning. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 341–352, 1992.

Kuchibhotla, A. K. and Berk, R. A. Nested conformal prediction sets for classification with applications to probation data. 2021. URL `https://arxiv.org/abs/2104.09358`.

Lei, J. and Wasserman, L. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):71–96, 2014.

Lindemann, L., Cleaveland, M., Shim, G., and Pappas, G. J. Safe planning in dynamic environments using conformal prediction. 2022. doi: 10.48550/ARXIV.2210.10254. URL `https://arxiv.org/abs/2210.10254`.

Lu, C., Lemay, A., Chang, K., Höbel, K., and Kalpathy-Cramer, J. Fair conformal predictors for applications in medical imaging. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12008–12016, Jun. 2022. doi: 10.1609/aaai.v36i11.21459. URL `https://ojs.aaai.org/index.php/AAAI/article/view/21459`.

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.

Papadopoulos, H., Proedrou, K., Vovk, V., and Gammerman, A. Inductive confidence machines for regression. In *European Conference on Machine Learning*, pp. 345–356. Springer, 2002.

Romano, Y., Barber, R. F., Sabatti, C., and Candès, E. With Malice Toward None: Assessing Uncertainty via Equalized Coverage. *Harvard Data Science Review*, 2(2), apr 30 2020. https://hdsr.mitpress.mit.edu/pub/qedrwcz3.

Rothblum, G. N. and Yona, G. Multi-group agnostic pac learnability. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 9107–9115. PMLR, 2021.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks for group shifts: On the import ance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020.

Shalev-Shwartz, S. and Ben-David, S. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014. ISBN 978-1-10-705713-5.

Tosh, C. J. and Hsu, D. Simple and near-optimal algorithms for hidden stratification and multi-group learning. In *International Conference on Machine Learning*, pp. 21633–21657. PMLR, 2022.

Vovk, V. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pp. 475–490. PMLR, 2012.

Vovk, V., Gammerman, A., and Shafer, G. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005.

A Multi-group Learning with Hierarchical Structure

Algorithm 1 in Section 4.2 is a special case of Algorithm 2 stated below, and its correctness and validity guarantees (Theorem 4.1 and Corollary 4.2) follow directly from the correctness and guarantees of Algorithm 2. In this section, we include Algorithm 2 in full generality in the context of multi-group learning. We note that this algorithm may be of independent interest to the multi-group learning literature for its "near-optimal" rates and simple resulting predictor.

In Section A.1, we restate the problem setting of multi-group learning, introduced earlier in Section 3.1, and we briefly discuss Prepend Tosh & Hsu [2022], the multi-group learning algorithm that `MGL-Tree` (Algorithm 2) extends. After stating some lemmas necessary for our following proofs in Section A.2, we state `MGL-Tree` (Algorithm 2) in Section A.3 and prove its correctness and multi-group learning guarantees in Section A.4.

A.1 Multi-group Learning and Prepend

We will eventually present and analyze Algorithm 2 (`MGL-Tree`) in the standard statistical learning setting with the aim of achieving the multi-group (agnostic) learning guarantees introduced by Rothblum & Yona [2021]. To do so, we first re-introduce some notation for multi-group learning, reminiscent of the brief discussion in Section 3.1.

Let $\mathcal{X}$ be an input space, $\mathcal{Y}$ a label space, and $\mathcal{A}$ a prediction space. Let $\mathbb{P} \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$ be a joint distribution. Let $\ell : \mathcal{A} \times \mathcal{Y} \to [0, 1]$ be a loss function. Let $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^n$ be a dataset of n i.i.d. examples drawn from $\mathbb{P}$. $\mathcal{G}$ is a collection of possibly overlapping groups, which are just subsets of the input space $\mathcal{X}$. Let the groups $g \in \mathcal{G}$ be defined as in Section 2.1. For now, we defer the definition of a *hierarchically structured* collection of groups $\mathcal{G}$ to Section A.2.

For any predictor, $f : \mathcal{X} \to \mathcal{A}$, the *group conditional risk* of f on group g is:

$$L(f \mid g) := \mathbb{E}\left[\ell(f(X), Y) \mid X \in g\right],$$

where the conditional expectation is over $(X, Y) \sim \mathbb{P}_g$ and any internal randomization of f. Similarly, define the *group conditional empirical risk* of f on group g as:

$$L_{\mathcal{D}}(f \mid g) := \frac{\sum_{i \in [n]} g(X_i)\ell(f(X_i), Y_i)}{\sum_{i \in [n]} g(X_i)}.$$

We will refer to a "benchmark" hypothesis class as some set of predictor functions $\mathcal{H} \subseteq \mathcal{A}^{\mathcal{X}}$. Here, "benchmark" is meant to indicate that the resulting predictor from a multi-group learning algorithm need not be contained in $\mathcal{H}$; in fact, this is necessary for multi-group learning in general.

The goal in multi-group learning is to find an algorithm such that, given a benchmark hypothesis class $\mathcal{H}$ and dataset $\mathcal{D}$, outputs a predictor f such that excess group conditional risks are simultaneously bounded for all $g \in \mathcal{G}$:

$$L(f \mid g) \leq \inf_{h \in \mathcal{H}} L(h \mid g) + \epsilon(\mathcal{D}, g, \delta), \tag{7}$$

with probability $1 - \delta$. $\epsilon(\mathcal{D}, g, \delta)$ is the *excess error* of f to the best hypothesis in $\mathcal{H}$ for group g.

The Prepend algorithm of Tosh & Hsu [2022] runs with finite $\mathcal{H}$ and $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^n$ i.i.d. training examples to produce a multi-group learning guarantee guarantee of

$$L(f \mid g) \leq \min_{h \in \mathcal{H}} L(h \mid g) + O\left(\sqrt[3]{\frac{\log |\mathcal{H}||\mathcal{G}|}{\gamma_n n_g}}\right) \quad \text{for all } g \in \mathcal{G},$$

where $\gamma_n := \min_{g \in \mathcal{G}} n_g/n$ and $n_g := |\{i \in [n] : X_i \in g\}|$. If we knew the group $g \in \mathcal{G}$ beforehand, then Empirical Risk Minimization (ERM) applied to the n_g examples in group g would yield the excess error rate $O(\sqrt{\log|\mathcal{H}|/n_g})$ Shalev-Shwartz & Ben-David [2014], which suggests that Prepend is suboptimal, in both its exponent and its denominator. However, by assuming that $\mathcal{G}$ is hierarchically structured, Algorithm 2 achieves a rate competitive to this by incurring an extra $\log|\mathcal{G}|$ factor: $O(\sqrt{\log|\mathcal{H}||\mathcal{G}|/n_g})$. In addition, Algorithm 2 retains the simplicity of Prepend – its output is a simple decision tree made up of hypotheses in $\mathcal{H}$. This gives it a qualitative advantage over the other complex, randomized multi-group learning predictor presented in Tosh & Hsu [2022] while achieving the same "near-optimal" excess error $O(\sqrt{\log|\mathcal{H}||\mathcal{G}|/n_g})$.

A.2 Lemmas for Multi-group Learning with Hierarchically Structured Groups

In order to achieve the near-optimal rate and simpler predictor output by Algorithm 2, we will impose the natural restriction that $\mathcal{G}$ is *hierarchically structured*. These definitions were introduced in Section 4.1, and we use the same terminology and notation here. We introduce several key lemmas that the proofs in Section A.4 depends on.

First, we present two lemmas that allow us to decompose the group conditional risk and group conditional empirical risk when we condition on a union of disjoint groups. We will apply the following two lemmas to the leaf nodes of the tree given by a hierarchically structured $\mathcal{G}$.

Lemma A.1 (Disjoint Decomposition of Group Conditional Empirical Risk)**.** *Let $g_1, \dots g_N$ be N disjoint groups, and let $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^n$ 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 $\bigcup_{k=1}^N g_k$ decomposes:*

$$L_{\mathcal{D}}(f \mid \bigcup_{k=1}^N g_k) = \sum_{k=1}^N \frac{n_{g_k}}{n_{\cup g}} L_{\mathcal{D}}(f \mid g_k). \tag{8}$$

Proof. By definition of group conditional empirical risk,

$$L_{\mathcal{D}}(f \mid \bigcup_{k=1}^N g_k) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\left\{X \in \bigcup_{k=1}^N g_k\right\} \ell(f(X_i), Y_i).$$

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

$$\begin{aligned}
\frac{1}{n} \sum_{i=1}^n \mathbb{1}\left\{X \in \bigcup_{k=1}^N g_k\right\} \ell(f(X_i), Y_i) &= \frac{1}{n_{\cup g}} \sum_{i=1}^n \left(\sum_{k=1}^N g_k(X) \ell(f(X_i), Y_i)\right) \\
&= \frac{1}{n_{\cup g}} \sum_{k=1}^N \left(\sum_{i=1}^n g_k(X) \ell(f(X_i), Y_i)\right) \\
&= \frac{1}{n_{\cup g}} \sum_{k=1}^N n_{g_k} L_{\mathcal{D}}(f \mid g_k) = \sum_{k=1}^N \frac{n_{g_k}}{n_{\cup g}} L_{\mathcal{D}}(f \mid g_k).
\end{aligned}$$

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

Lemma A.2 (Disjoint Decomposition of Group Conditional Risk)**.** *Let $g_1, \dots g_N$ be N disjoint groups. Then, the group conditional risk for $\bigcup_{k=1}^N g_k$ decomposes:*

$$L(f \mid \bigcup_{k=1}^{N} g_k) = \sum_{k=1}^{N} \frac{\Pr\{X \in g_k\}}{\sum_{j=1}^N \Pr\{X \in g_j\}} L(f \mid g_k). \tag{9}$$

Proof. By definition of group conditional risk,

$$L(f \mid g) = \mathbb{E}\left[\ell(f(X), Y) \,|\, X \in g\right].$$

We first claim that $L(f \mid g) = \frac{1}{\Pr\{X \in g\}} \mathbb{E}\left[g(X)\ell(f(X), Y)\right]$. This follows from:

$$\begin{aligned}
\mathbb{E}\left[g(X)\ell(f(X), Y)\right] &= \mathbb{E}\left[\mathbb{E}\left[g(X)\ell(f(X), Y) \,|\, X \in g\right]\right] \\
&= \mathbb{E}\left[g(X)\mathbb{E}\left[\ell(f(X), Y) \,|\, X \in g\right]\right] \\
&= \Pr\{X \in g\}\mathbb{E}\left[\ell(f(X), Y) \,|\, X \in g\right] \\
&= \Pr\{X \in g\} L(f \mid g).
\end{aligned}$$

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

$$L(f \mid \bigcup_{k=1}^{N} g_k) = \frac{1}{\Pr\{X \in \bigcup_{k=1}^N g_k\}} \mathbb{E}\left[\mathbb{1}\left\{X \in \bigcup_{k=1}^{N} g_k\right\} \ell(f(X), Y)\right].$$

Because $g_1, \dots, g_N$ are disjoint, we can use additivity:

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

Using the same fact that $\Pr\{X \in g_k\} L(f \mid g_k) = \mathbb{E}\left[g_k(X)\ell(f(X), Y)\right]$, we get the desired result:

$$= \sum_{k=1}^{N} \frac{\Pr\{X \in g_k\}}{\sum_{j=1}^N \Pr\{X \in g_j\}} L(f \mid g_k).$$

□

The next lemma is from Tosh & Hsu [2022], but we restate it here for convenience. This is a uniform convergence result for group conditional risks. We'll use its immediate consequence for hierarchically structured $\mathcal{G}$, Lemma A.4, to prove excess error bounds. First, we define a *shattering coefficient* of a set of functions. For a class of $\{0, 1\}$-valued functions $\mathcal{F}$ defined over a domain $\mathcal{X}$, the k-th shattering coefficient is given by:

$$\Pi_k(\mathcal{F}) := \max_{x_1, \dots, x_k \in \mathcal{X}} |\{(f(x_1), \dots, f(x_k)) : f \in \mathcal{F}\}|.$$

We also define $((\ell \circ \mathcal{H})_{\text{thresh}}$ as the thresholded, loss-composed class for ℓ and $\mathcal{H}$, which is the class of functions

$$(\ell \circ \mathcal{H})_{\text{thresh}} := \{(x, y) \mapsto \mathbf{1}\{(\ell(h(x), y))\} > \tau : h \in \mathcal{H}, \tau \in \mathbb{R}\}.$$

Now, we can restate the result for uniform convergence on group conditional risks, which is Theorem 1 from Tosh & Hsu [2022].

Lemma A.3 (Theorem 1 in Tosh & Hsu [2022]). *Let $\mathcal{H}$ be a hypothesis class and let $\mathcal{G}$ be a set of groups. Given a dataset $\mathcal{D}$ of n i.i.d. examples, with probability at least $1-\delta$,*

$$|L(h \mid g) - L_{\mathcal{D}}(h \mid g)| \leq \min\left\{9\sqrt{\frac{D}{n_g}}, 7\sqrt{\frac{DL_{\mathcal{D}}(h \mid g)}{n_g} + \frac{16D}{n_g}}\right\}, \tag{10}$$

for all $h \in \mathcal{H}$ and $g \in \mathcal{G}$, where $D = 2\log(\Pi_{2n}((\ell \circ \mathcal{H})_{\text{thresh}})\Pi_{2n}(\mathcal{G})) + \log(8/\delta)$.

We can simplify the D term to get the following conditional uniform convergence bounds. Using the crudest bounds on the terms in D (which will be sufficient), we immediately obtain the following useful lemma for group conditional uniform convergence. This is the lemma we will use to prove excess error bounds in the sequel.

Lemma A.4 (Group Conditional Uniform Convergence). *Let $\mathcal{H}$ be a hypothesis class and let $\mathcal{G}$ be a set of groups. Given a dataset $\mathcal{D}$ of n i.i.d. examples, with probability at least $1-\delta$,*

$$|L(h|g) - L_{\mathcal{D}}(h|g)| \leq 9\sqrt{\frac{2\log(|\mathcal{G}||\mathcal{H}|) + \log(8/\delta)}{n_g}}, \tag{11}$$

for all $h \in \mathcal{H}$ and all $g \in \mathcal{G}$. For $\mathcal{H}$ with VC-dimension bounded by $d > 0$, with probability at least $1-\delta$,

$$|L(h|g) - L_{\mathcal{D}}(h|g)| \leq 9\sqrt{\frac{2\log(|\mathcal{G}|(4n)^d) + \log(8/\delta)}{n_g}}, \tag{12}$$

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

A.3 `MGL-Tree` Algorithm

We now present `MGL-Tree` (Algorithm 2). This algorithm outputs a final decision tree predictor, $\hat{f} : \mathcal{X} \to \mathcal{A}$. Each node of the decision tree is a group $g \in \mathcal{G}$ with an associated *working predictor* $\hat{h}_w^g : \mathcal{X} \to \mathcal{A}$. For each group $g \in \mathcal{G}$, we'll denote $\hat{h}_g$ to be the ERM minimizer of group conditional empirical risk:

$$\hat{h}_g := \operatorname*{arg\,min}_{h \in \mathcal{H}} L_{\mathcal{D}}(h \mid g) = \operatorname*{arg\,min}_{h \in \mathcal{H}} \frac{\sum_{i \in [n]} g(X_i)\ell(h(X_i), Y_i)}{\sum_{i \in [n]} g(X_i)}.$$

We construct the decision tree as follows. First, we generate the tree in the way described in Section 4 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 $\hat{h}_w^{\mathcal{X}} \leftarrow \hat{h}^{\mathcal{X}}$, the ERM minimizer of group conditional empirical risk for all of $\mathcal{X}$ – this is just standard empirical risk $L_{\mathcal{D}}(h)$.

To assign working predictors $\hat{h}_w^g$ 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 $\hat{h}_w^{\text{pa}(g)}$ denote the working predictor for the parent of node g. The main idea is to set the working predictor at node g to $\hat{h}_w^g \leftarrow \hat{h}^g$ 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: $\hat{h}_w^g \leftarrow \hat{h}_w^{\text{pa}(g)}$. To show that Algorithm 2 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. This is detailed formally in Algorithm 2.

Algorithm 2 `MGL-Tree`

Input:

1: $\mathcal{D}$, 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 $g \in \mathcal{G}$

Output: Decision tree $\hat{f} : \mathcal{X} \to \mathcal{A}$.

4: Order $\mathcal{G}$ into a tree, where g is an ancestor of g' if $g' \subseteq g$.
5: Initialize the root working predictor: $\hat{h}_w^{\mathcal{X}} \leftarrow \hat{h}^{\mathcal{X}}$.
6: **for** each node $g \in \mathcal{G} \setminus \{\mathcal{X}\}$ in breadth-first order **do**
7: Compute:

$$\mathrm{err}_g := \frac{1}{|\mathcal{D}_g|} \sum_{(X,Y)\in\mathcal{D}} g(X) \left(\ell\left(\hat{h}_w^g(X), Y\right) - \ell\left(\hat{h}_g(X), Y\right)\right) - \epsilon_n(g)$$

8: **if** $\mathrm{err}_g \geq 0$ **then**
9: Set $\hat{h}_w^g \leftarrow \hat{h}^g$.
10: **else**
11: Set $\hat{h}_w^g \leftarrow \hat{h}_w^{\mathrm{pa}(g)}$.
12: **end if**
13: **end for**
14: **return** $\hat{f} : \mathcal{X} \to \mathcal{A}$, a decision tree predictor.

To evaluate using this decision tree $\hat{f}$, we simply propagate a new example X down the tree, starting from the root, finding the "deepest" node g^* in the tree that still contains X. We move from a parent to a child whenever the child contains X. The final output uses the working predictor at this "deepest" node g^*: $\hat{f}(X) := \hat{h}_w^{g^*}(X)$.

A.4 `MGL-Tree` Analysis and Sample Complexity

In this section, we prove the correctness and multi-group learnability guarantees of Algorithm 2. We need a bit of terminology to denote the change at an update operation. Let $\hat{f}_{\mathrm{old}}$ denote the state of the decision tree *before* an update operation. In Algorithm 2, this corresponds to the state of the decision tree at line 7 in each iteration of the main BFS loop. We will use $\hat{f}_{\mathrm{old}}$ extensively in the proofs of Theorem A.6 and Theorem A.7.

We state one more obvious lemma concerning the behavior of $\hat{f}$ when we choose to inherit the parent's working predictor, $\hat{h}_w^{\mathrm{pa}(g)}$. When we do this, $\hat{f}$ is functionally equivalent to $\hat{f}_{\mathrm{old}}$ on group g and all the nodes on the path from g back up to the root.

Lemma A.5 (Behavior of $\hat{f}_{\mathrm{old}}$)**.** *Consider any step of Algorithm 2 where we are considering $g \in \mathcal{G}$. Let $\hat{f}_{\mathrm{old}}$ be the decision tree at this step before updating (the state of the tree at line 7). Let $\hat{h}_g \in \arg\min_{h\in\mathcal{H}} L_{\mathcal{D}}(h \mid g)$ for all $g \in \mathcal{G}$. Then, for all $X \in g$, $\hat{f}_{\mathrm{old}}(X) = h_{g'}(X)$ for some $g' \supset g$ already visited by the algorithm.*

Proof. This just follows by induction. For the first step of Algorithm 2, $\hat{f}_{\mathrm{old}}$ is simply $h \in \arg\min_{h\in\mathcal{H}} L_{\mathcal{D}}(h)$, 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

$X \in g$, then, by hierarchical structure and BFS, $X \in g'$ for some $g \subset g'$ because we've visited all parents before their children. Therefore, $\hat{f}_{\text{old}}$ uses $h_{g'}$ for some $g' \supset g$. □

Lemma A.5 allows us to apply Lemma A.4, our conditional uniform convergence bound, on $\hat{f}_{\text{old}}$, as it is functionally equivalent to some $h \in \mathcal{H}$, our benchmark hypothesis class.

The key to Algorithm 2 is that each update operation at any node g does not make $\hat{f}$ violate the error bounds it satisfied further up the tree. We observe that, at an update iteration (when $\text{err}_g \geq 0$), either we accept the (conditional) ERM predictor $\hat{h}^g_w \leftarrow \hat{h}^g$ or we inherit the parent's working predictor $\hat{h}^g_w \leftarrow \hat{h}^{\text{pa}(g)}_w$. See Figure 1.

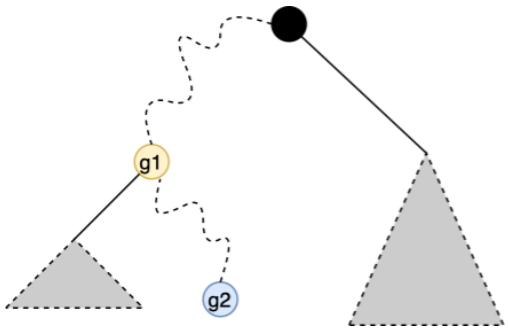

Figure 1: Let g_1, the yellow node, be a group that Algorithm 2 has already dealt with. Suppose $\hat{f}$ updates on g_2. We see that g_1 is on the path from g_2 to the root. We need to show that the inequality for g_1 is not violated after the update.

Theorem A.6 (Correctness of `MGL-Tree`)**.** *Let $\mathcal{G}$ be a hierarchically structured collection of groups, let $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^n$ be i.i.d. training data drawn from any distribution $\mathbb{P} \in \mathcal{P}(\mathcal{X} \times \mathcal{Y})$, and let $\epsilon_n : \mathcal{G} \to (0, 1)$ be any error rate function. Then, Algorithm 2 run on these parameters outputs a predictor $\hat{f} : \mathcal{X} \to \mathcal{Y}'$ satisfying:*

$$L_{\mathcal{D}}(\hat{f}|g) \leq \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h|g) + \epsilon_n(g), \quad \text{for all } g \in \mathcal{G}. \tag{13}$$

Proof. Consider any $g^* \in \mathcal{G}$, which corresponds to a node in the decision tree, $\hat{f}$. 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 (13) 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 (13) 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 $\hat{f} \leftarrow \hat{f}_{\text{old}}$ from the previous round, and we are done. All inequalities previously satisfied must continue to be satisfied because $\hat{f}$ 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 $g \notin P_{g^*}$, $\hat{f}$ continues to satisfy (13) because $g \cap g^* = \emptyset$, so for $X \in g$, $\hat{f}(X) = \hat{f}_{\text{old}}(X)$, 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 $\hat{f}$ 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 $L_{\mathcal{D}}(\hat{f} \mid \hat{g}_j) \leq L_{\mathcal{D}}(\hat{f}_{\text{old}} \mid \hat{g}_j)$. Denote $\overline{g}$ as the complement of g, and apply Lemma A.1:

$$\begin{aligned}
L_{\mathcal{D}}(\hat{f} \mid \hat{g}_j) &= L_{\mathcal{D}}(\hat{f} \mid (\hat{g}_j \cap g^*) \cup (\hat{g}_j \cap \overline{g^*})) \\
&= L_{\mathcal{D}}(\hat{f} \mid \hat{g}_j \cap g^*) \frac{\Pr\{X \in \hat{g}_j \cap g^*\}}{\Pr\{X \in \hat{g}_j\}} + L_{\mathcal{D}}(\hat{f} \mid \hat{g}_j \cap \overline{g^*}) \frac{\Pr\{X \in (\hat{g}_j \cap \overline{g^*})\}}{\Pr\{X \in \hat{g}_j\}} \\
&= L_{\mathcal{D}}(\hat{f} \mid g^*) \Pr\{X \in g^* \mid X \in \hat{g}_j\} + L_{\mathcal{D}}(\hat{f} \mid \hat{g}_j \cap \overline{g^*}) \Pr\{X \in \overline{g^*} \mid X \in \hat{g}_j\} \\
&= L_{\mathcal{D}} \Pr\{X \in g^* \mid X \in \hat{g}_j\} + L_{\mathcal{D}}(\hat{f}_{\text{old}} | \hat{g}_j \cap \overline{g^*}) \Pr\{X \in \overline{g^*} \mid X \in \hat{g}_j\}. \qquad (14)
\end{aligned}$$

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

$$L_{\mathcal{D}}(h^* \mid g^*) \leq L_{\mathcal{D}}(\hat{f}_{\text{old}} \mid g^*), \qquad (15)$$

so combining (14) and (15),

$$\begin{aligned}
L_{\mathcal{D}}(\hat{f} \mid \hat{g}_j) &= L_{\mathcal{D}}(h^* \mid g^*) \Pr\{X \in g^* \mid X \in \hat{g}_j\} + L_{\mathcal{D}}(\hat{f}_{\text{old}} \mid \hat{g}_j \cap \overline{g^*}) \Pr\{X \in \overline{g^*} \mid X \in \hat{g}_j\} \\
&\leq L_{\mathcal{D}}(\hat{f}_{\text{old}} \mid g^*) \Pr\{X \in g^* \mid X \in \hat{g}_j\} + L_{\mathcal{D}}(\hat{f}_{\text{old}} \mid \hat{g}_j \cap \overline{g^*}) \Pr\{X \in \overline{g^*} \mid X \in \hat{g}_j\} \\
&= L_{\mathcal{D}}(\hat{f}_{\text{old}} \mid \hat{g}_j) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h \mid \hat{g}_j) + \epsilon_n(\hat{g}_j), \qquad (16)
\end{aligned}$$

where the final equality in (16) follows from another application of Lemma A.1. Because

$$L_{\mathcal{D}}(\hat{f} \mid \hat{g}_j) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h \mid \hat{g}_j) + \epsilon_n(\hat{g}_j), \qquad (17)$$

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

We may now state and prove the main theorem for the multi-group learning guarantee of `MGL-Tree`.

Theorem A.7 (`MGL-Tree` Sample Complexity)**.** *Suppose $\mathcal{H}$ is a benchmark hypothesis class, $\mathcal{G}$ is a hierarchically structured collection of groups and $\mathcal{D} = \{(X_i, Y_i)\}_{i=1}^n$ is the i.i.d. training dataset input to Algorithm 2. For finite $\mathcal{H}$, if we run Algorithm 2 to completion, with*

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

then it will output a decision tree $\hat{f}$ that, with probability $1-\delta$ over the n training examples, satisfies:

$$L(\hat{f} \mid g) \leq \inf_{h \in \mathcal{H}} L(h \mid g) + 36\sqrt{\frac{2\log(|\mathcal{G}||\mathcal{H}|) + \log(8/\delta)}{n_g}} \quad \text{for all } g \in \mathcal{G}. \qquad (18)$$

For $\mathcal{H}$ with VC dimension $d > 0$, if we run Algorithm 2 to completion, with

$$\epsilon_n(g) := 18\sqrt{\frac{2(d+1)\log(4n) + \log(8/\delta)}{n_g}},$$

then we obtain:

$$L(\hat{f} \mid g) \leq \inf_{h \in \mathcal{H}} L(h \mid g) + 36\sqrt{\frac{2(d+1)\log(4n) + \log(8/\delta)}{n_g}} \quad \textit{for all } g \in \mathcal{G}. \tag{19}$$

Proof. We will show this by induction on each iteration (visited group g) of Algorithm 2. Condition on the event that we drew our i.i.d. dataset of size n and Equation (11) from Lemma A.4 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 $h \in \mathcal{H}$:

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

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 $\hat{f}$ satisfies

$$L(\hat{f} \mid g) \leq \min_{h \in \mathcal{H}} L(h \mid g) + 4UC(g). \tag{20}$$

For the base case, we just need to show that our starting decision tree $\hat{f}$, where each node is initialized with $h_0 \in \arg\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$, satisfies inequality (20) for $g = \mathcal{X}$. Note that this is just standard unconditional risk. This is immediately true from uniform convergence and $\hat{f}$ using the ERM predictor h_0 Shalev-Shwartz & Ben-David [2014], so

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

For the inductive hypothesis, assume that we are on the step of the BFS in Algorithm 2 concerned with $g \in \mathcal{G}$. Denote $\mathcal{G}'$ as the nodes that we already visited in our BFS, and $\hat{f}_{\text{old}}$ as the decision list *before* the possible update, as before. Then,

$$L(\hat{f}_{\text{old}} \mid g') \leq \min_{h \in \mathcal{H}} L(h \mid g') + 4UC(g') \tag{21}$$

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

To prove the induction, we aim to show that (20) is true for our current iteration's node g as well, regardless of whether we updated $\hat{f}$ or not. Let $\hat{f}$ be the decision list *after* the update step of Algorithm 2, and let $\hat{h}_g \in \arg\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h \mid g)$. We want to show, for all $h \in \mathcal{H}$:

$$L(\hat{f}|g) - L(h|g) \leq 4UC(g) \tag{22}$$

$$L(\hat{f}|g') - L(h|g') \leq 4UC(g'), \text{ for all } g' \in \mathcal{G}'. \tag{23}$$

So, showing (22) and (23) are our goals for each of our two cases: whether we update or not. These two cases depend on whether $L_{\mathcal{D}}(\hat{f}_{\text{old}}|g) - L_{\mathcal{D}}(h_g|g) \leq \epsilon_n(g)$ or $L_{\mathcal{D}}(\hat{f}_{\text{old}}|g) - L_{\mathcal{D}}(h_g|g) > \epsilon_n(g)$, 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 $L_{\mathcal{D}}(\hat{f}_{\text{old}}|g) - L_{\mathcal{D}}(h_g|g) \leq \epsilon_n(g)$. Then, because we do not update, $\hat{f} = \hat{f}_{\text{old}}$, so (23) is immediately fulfilled

because $\hat{f}$ is functionally equivalent to $\hat{f}_{\text{old}}$, which, by induction satisfied the inequality already for all $g' \in \mathcal{G}'$. It suffices to show (22) for this case. Fix any $h \in \mathcal{H}$. First, with Lemma A.5, we can apply conditional uniform convergence on $\hat{f} = \hat{f}_{\text{old}}$. Then,

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

The first inequality is from Lemma A.5 and Lemma A.4. The third inequality is from the fact that $\hat{h}_g$ is the optimal ERM predictor conditioned on $X \in g$. This proves (22) for the first case where we do not update $\hat{f}$.

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

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

The first equality is because $\hat{f}$ is functionally equivalent to $\hat{h}_g$ on all $X \in g$, and the first inequality comes from applying Lemma A.4 twice to get sample risk for h and $\hat{h}_g$. This proves (22). It suffices to prove (23). 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: $\hat{f} = \hat{f}_{\text{old}}$. Then, for any $h \in \mathcal{H}$, we maintain the same guarantee we had before, fulfilling (23) for all g'_{np}:

$$L(\hat{f} \mid g'_{np}) - L(h \mid g'_{np}) = L(\hat{f}_{\text{old}} \mid g'_{np}) - L(h \mid g'_{np}) \leq 4UC(g'_{np}).$$

To finish the proof, it suffices to show that (23) 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.2:

$$\begin{aligned} L(\hat{f} \mid g'_p) &= L(\hat{f} \mid (g'_p \cap g) \cup (g'_p - g)) \\ &= \frac{\Pr\{X \in (g'_p \cap g)\}}{\Pr\{X \in g'_p\}} L(\hat{f} \mid g'_p \cap g) + \frac{\Pr\{X \in (g'_p - g)\}}{\Pr\{X \in g'_p\}} L(\hat{f} \mid g'_p - g) \\ &= \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} L(\hat{f} \mid g) + \frac{\Pr\{X \in (g'_p - g)\}}{\Pr\{X \in g'_p\}} L(\hat{f} \mid g'_p - g). \end{aligned}$$

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

$$L(\hat{f} \mid g'_p) = \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} L(\hat{h}_g \mid g) + \frac{\Pr\{X \in (g'_p - g)\}}{\Pr\{X \in g'_p\}} L(\hat{f}_{\text{old}} \mid g'_p - g).$$

Apply Lemma A.4 for conditional uniform convergence on $\hat{h}_g$:

$$L(\hat{f} \mid g'_p) \leq \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} L_{\mathcal{D}}(\hat{h}_g \mid g) + \frac{\Pr\{X \in (g'_p - g)\}}{\Pr\{X \in g'_p\}} L(\hat{f}_{\text{old}} \mid g'_p - g) + \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} UC(g).$$

Adding and subtracting $\epsilon_n(g)$ to use the fact that we are in the update case where $L_{\mathcal{D}}(\hat{f}_{\text{old}}|g) > L_{\mathcal{D}}(h_g|g) + \epsilon_n(g)$:

$$\begin{aligned}
&= \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} L_{\mathcal{D}}(h_g \mid g) + \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} \epsilon_n(g) \\
&\quad + \frac{\Pr\{X \in (g'_p - g)\}}{\Pr\{X \in g'_p\}} L(\hat{f}_{\text{old}} \mid g'_p - g) + \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} UC(g) - \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} \epsilon_n(g) \\
&\leq \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} L_{\mathcal{D}}(\hat{f}_{\text{old}} \mid g) + \frac{\Pr\{X \in (g'_p - g)\}}{\Pr\{X \in g'_p\}} L(\hat{f}_{\text{old}} \mid g'_p - g) + \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} UC(g) - \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} \epsilon_n(g).
\end{aligned}$$

Finally, using Lemma A.5 on $\hat{f}_{\text{old}}$ on $X \in g$, applying Lemma A.4 again, and recombining terms with Lemma A.2,

$$\begin{aligned}
&\leq \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} L(\hat{f}_{\text{old}} \mid g) + \frac{\Pr\{X \in (g'_p - g)\}}{\Pr\{X \in g'_p\}} L(\hat{f}_{\text{old}} \mid g'_p - g) + \frac{2\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} UC(g) - \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} \epsilon_n(g) \\
&= L(\hat{f}_{\text{old}} \mid g'_p) + \frac{2\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} UC(g) - \frac{\Pr\{X \in g\}}{\Pr\{X \in g'_p\}} \epsilon_n(g) \leq L(h \mid g'_p) + 4UC(g'_p).
\end{aligned}$$

The final line follows by our choice of $\epsilon_n(g) = 2UC(g)$ and the inductive hypothesis on g'_p. This shows (23) for the second case where we update $\hat{f}$, and thus completes our proof.

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

$$UC(g) = 9\sqrt{\frac{2(d+1)\log(4n) + \log(8/\delta)}{n_g}}$$

and $\epsilon_n(g) = 2UC(g)$. □

A.5 Proof of Theorem 4.1 and Corollary 4.2

In this section, we briefly demonstrate how Algorithm 1 in Section 4 is a special case of Algorithm 2 and prove its validity guarantee. For Algorithm 1 instantiated with ICP, we show that this follows almost immediately from Theorem A.7 and the training conditional validity guarantee Proposition 2.1 (restated from Proposition 2a from Vovk [2012]). Recall that, by Proposition 2.1, for $\alpha, \delta \in (0, 1)$ and a dataset $\mathcal{D} = \mathcal{D}^{(\text{tr})} \cup \mathcal{D}^{(\text{cal})}$ running ICP produces a set predictor C which satisfies

$$\Pr\left[Y \notin C(X) \,|\, \mathcal{D}\right] \leq \alpha + \sqrt{\frac{\log(1/\delta)}{2|\mathcal{D}^{(\text{cal})}|}}, \tag{24}$$

for newly drawn random example (X, Y), with probability $1 - \delta$ over $\mathcal{D}$. Let the training dataset for Algorithm 2 be $\mathcal{D}^{(\text{mg})}$, an i.i.d. dataset independent from $\mathcal{D}$, throughout this section. We can now state the reduction for Theorem 4.1 and Corollary 4.2.

Proof. Run Algorithm 2 with the following setup, and we obtain Algorithm 1. Let $\mathcal{A} = 2^{\mathcal{Y}}$, so the decision tree is a set predictor $C : \mathcal{X} \to 2^{\mathcal{Y}}$. Each working predictor for node $g \in \mathcal{G}$ is also a set predictor. Let the loss function be the 0/1 miscoverge loss (Section 3.2) that checks whether Y is in some subset of $2^{\mathcal{Y}}$; for an example (X, Y) and set predictor C, the loss is:

$$\ell(Y, C(X)) := \mathbb{1}\{Y \notin C(X)\}. \tag{25}$$

We run Algorithm 1 with $\mathcal{D}^{(\text{mg})}$. In Algorithm 1, we are also provided a benchmark hypothesis class $\hat{\mathcal{H}} = \{C^g : g \in \mathcal{G}\}$ of set predictors $C^g : \mathcal{X} \to 2^{\mathcal{Y}}$ that are (α_g, δ)-group conditionally valid for $\mathbb{P}_g$. Taking a conditional expectation of (25), we obtain the group conditional risk

$$L(C \mid g) := \mathbb{E}\left[\ell(Y, C(X)) \mid X \in g\right] = \Pr\left[Y \notin C(X) \,\middle|\, \mathcal{D}_g^{(\text{mg})}, X \in g\right],$$

which is exactly the quantity we want to upper bound to obtain group conditional validity. Apply Threorem A.7 to immediately obtain Theorem 4.1, the $((\alpha'_g : g \in \mathcal{G}), \delta)$-group conditional validity for $(\mathbb{P}, \mathcal{G})$ of the decision tree set predictor C, with:

$$\alpha'_g = \alpha_g + 36\sqrt{\frac{4\log(|\mathcal{G}|) + \log(8/\delta)}{|\mathcal{D}_g^{(\text{mg})}|}} \quad \text{for all } g \in \mathcal{G}.$$

This proves Theorem 4.1.

Corollary 4.2 just follows from Theorem 4.2 by constructing $\hat{\mathcal{H}}$ from ICP applied on $\mathcal{D} = \mathcal{D}^{(\text{tr})} \cup \mathcal{D}^{(\text{cal})}$. Using the guarantee from Proposition 2.1, for each $g \in \mathcal{G}$, we run ICP using $\mathcal{D}_g$ (the dataset with *only* examples in g) at level $1 - \alpha$ to obtain set predictors $C^g : \mathcal{X} \to 2^{\mathcal{Y}}$ with the guarantee:

$$\Pr\left[Y \notin C^g(X) \mid \mathcal{D}_g, X \in g\right] \leq \alpha + \sqrt{\frac{\log(1/\delta)}{2|\mathcal{D}_g^{(\text{cal})}|}}. \tag{26}$$

These set predictors C^g form our finite benchmark hypothesis class $\hat{\mathcal{H}} = \{C_g : g \in \mathcal{G}\}$. A union bound over $\mathcal{G}$ and combining with Theorem 4.1 gives us our result. □