Fair Risk Control: A Generalized Framework for Calibrating Multi-group Fairness Risks

Lujing Zhang[1] **Aaron Roth**[2] **Linjun Zhang**[3]

Abstract

This paper introduces a framework for post-processing machine learning models so that their predictions satisfy multi-group fairness guarantees. Based on the celebrated notion of multicalibration, we introduce $(\boldsymbol{s}, \mathcal{G}, \alpha)-$GMC (Generalized Multi-Dimensional Multicalibration) for multi-dimensional mappings $\boldsymbol{s}$, constraint set $\mathcal{G}$, and a pre-specified threshold level α. We propose associated algorithms to achieve this notion in general settings. This framework is then applied to diverse scenarios encompassing different fairness concerns, including false negative rate control in image segmentation, prediction set conditional uncertainty quantification in hierarchical classification, and de-biased text generation in language models. We conduct numerical studies on several datasets and tasks.

1. Introduction

A common theme across the fairness in machine learning literature is that some measure of *error* or *risk* should be equalized across sub-populations. Common measures evaluated across demographic groups include false positive and false negative rates (Hardt et al., 2016) and calibration error (Kleinberg et al., 2016; Chouldechova, 2017). Initial work in this line gave methods for equalizing different risk measures on disjoint groups. A second generation of work gave methods for equalizing measures of risk across groups even when the groups could intersect – e.g. for false positive and negative rates (Kearns et al., 2018), calibration error (Úrsula Hébert-Johnson et al., 2018), regret (Blum & Lykouris, 2019; Rothblum & Yona, 2021), prediction set coverage (Jung et al., 2021; 2022; Deng et al., 2023), among other risk measures. In general, distinct algorithms are derived for each of these settings, and they are generally limited to one-dimensional predictors of various sorts.

In this work, we propose a unifying framework for fair risk control in settings with multi-dimensional outputs, based on multicalibration (Úrsula Hébert-Johnson et al., 2018). This framework is developed as an extension of the work by (Deng et al., 2023; Noarov & Roth, 2023), and addresses the need for calibrating multi-dimensional output functions. To illustrate the usefulness of this framework, we apply it to a variety of settings, including false negative rate control in image segmentation, prediction set conditional coverage guarantees in hierarchical classification, and de-biased text generation in language models. These applications make use of the additional power granted by our multi-dimensional extension of multicalibration.

1.1. Related Work

Multicalibration was introduced by (Úrsula Hébert-Johnson et al., 2018) as a fairness motivated constraint that informally asks that a 1-dimensional predictor of a binary-valued outcome be unbiased, conditional on both its own prediction and on membership of the input in some number of pre-defined groups (see also a line of prior work that asks for a similar set of guarantees under slightly different conditions (Dawid, 1985; Sandroni et al., 2003; Foster & Kakade, 2006)). Subsequently, multicalibration has been generalized in a number of ways. (Jung et al., 2021) generalizes multicalibration to real-valued outcomes, and defines and studies a variant of multicalibration that predicts variance and higher moments rather than means. (Gupta et al., 2022) extends the study of multicalibration of both means and moments to the online setting, and defines a variant of mulicalibration for quantiles, with applications to uncertainty estimation. (Bastani et al., 2022; Jung et al., 2022) gives more practical variants of quantile multicalibration with applications to conformal prediction, together with experimental evaluation. (Deng et al., 2023) gives an abstract generalization of 1-dimensional multicalibration, and show how to cast other algorithmic fairness desiderata like false positive rate control in this framework. (Noarov & Roth, 2023) gives a characterization of the scope of 1-dimensional multicalibration variants via a connection to property elici-

[*]Equal contribution [1]Peking University [2]University of Pennsylvania [3]Rutgers, University of New Jersey. Correspondence to: Linjun Zhang <linjun.zhang@rutgers.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

tation: informally, a property of a distribution can be multicalibrated if and only if it minimizes some 1-dimensional separable regression function. The primary point of departure of this paper is that we propose a multi-dimensional generalization of multicalibration: it can be viewed as the natural multi-dimensional generalization of (Deng et al., 2023).

Another line of work generalizes multicalibration in an orthogonal direction, leaving the outcomes binary valued but generalizing the class of checking rules that are applied. (Dwork et al., 2021) defines outcome indistinguishability, which generalizes multicalibration to require indistinguishability between the predicted and true label distributions with respect to a fixed but arbitrary set of distinguishers. (Foster & Hart, 2018) defines "smooth calibration" that relaxes calibration's conditioning event to be a smooth function of the prediction. (Gopalan et al., 2022) defines a hierarchy of relaxations called low-degree multicalibration that further relaxes smooth calibration and demonstrates desirable statistical properties. (Zhao et al., 2021) and (Noarov et al., 2023) define notions of calibration tailored to the objective function of a downstream decision maker. These last lines of work focus on multi-dimensional outputs.

These lines of work are part of a more general literature studying *multi-group fairness*. Work in this line aims e.g. to minimize disparities between false positive or false negative rates across groups (Kearns et al., 2018; 2019), or to minimize regret (measured in terms of accuracy) simultaneously across all groups (Blum & Lykouris, 2019; Rothblum & Yona, 2021; Globus-Harris et al., 2022; Tosh & Hsu, 2022). A common theme across these works is that the groups may be arbitrary and intersecting.

2. Notation

Let $\mathcal{X}$ represent a feature domain, $\mathcal{Y}$ represent a label domain, and $\mathcal{D}$ denote a joint (feature, label) data distribution. For a finite set A, we use $|A|$ and ΔA, to denote the cardinality of A and the simplex over A respectively. Specifically, $\Delta A = \{(p_1, p_2, \ldots, p_{|A|}) : 0 \le p_i \le 1, \sum_{i=1}^{|A|} p_i = 1\}$. Given a set $\mathcal{F}$, we use $\mathrm{Proj}_{\mathcal{F}}$ to denote the ℓ_2-projection onto the set.

We also introduce some shorthand notation. For two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ represents their inner product. For a positive integer T, we define $[T] = \{1, 2, \ldots, T\}$. For a function $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_m(\boldsymbol{x}))$, we denote $\|\boldsymbol{f}\|_\infty = \sup_{\boldsymbol{x} \in \mathcal{X}, i \in [m]} [f_i(\boldsymbol{x})]$.

3. Formulation and Algorithm

3.1. A generalized notion of Multicalibration

Let $\boldsymbol{x} \in \mathcal{X}$ represent the feature vector of the input, $\boldsymbol{y} \in \mathcal{Y}$ represent the label, and let $\boldsymbol{h}(\boldsymbol{x}) \in \mathcal{H}$ denote a multi-dimensional scoring function associated with the input. For example, in image segmentation tasks, $\boldsymbol{h}(\boldsymbol{x}) \in \mathbb{R}^k$ (k is the number of pixels) is intended to approximate the probability of a pixel being part of a relevant segment, often learned by a neural network. In text generation tasks, $\boldsymbol{h}(\boldsymbol{x})$ is the distribution over the vocabulary produced by a language model given context $\boldsymbol{x}$.

For $\boldsymbol{x} \in \mathcal{X}$, consider an objective function $\boldsymbol{f} : \mathcal{X} \to \mathcal{F} \subset \mathbb{R}^m$, defined as $\boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \ldots, f_m(\boldsymbol{x}))$, where $\mathcal{F}$ is a convex set. We denote the class of functions that $\boldsymbol{f}$ belongs to by $\mathcal{Q}$. For example, in text generation tasks, $\boldsymbol{f}(\boldsymbol{x})$ is the calibrated distribution over the output vocabulary and is multi-dimensional (with dimension equal to the vocabulary size); in binary classification tasks where h and f are both scalars, $f(\boldsymbol{x})$ is the threshold used to convert the raw score $h(\boldsymbol{x})$ into binary predictions, i.e. $\mathbb{1}_{\{h(\boldsymbol{x}) > f(\boldsymbol{x})\}}$.

We write $\boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}) : \mathcal{Q} \times \mathcal{X} \times \mathcal{H} \times \mathcal{Y} \times \mathcal{P} \to \mathbb{R}^l$ to denote a mapping functional of interest, where $\mathcal{D}$ is the joint distribution of $(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y})$ and $\mathcal{P}$ is the distribution space. Here, $\boldsymbol{s}$ is set to be a functional of $\boldsymbol{f}$ rather than a function of $\boldsymbol{f}(\boldsymbol{x})$, which offers us more flexibility that will be useful in our applications. For example, in text generation, where $\boldsymbol{h}(\boldsymbol{x}) \in \Delta\mathcal{Y}$ is the distribution over tokens output by an initial language model, our goal might be to find $\boldsymbol{f}(\boldsymbol{x}) \in \Delta\mathcal{Y}$, an adjusted distribution over tokens $y \in \mathcal{Y}$ with $|\mathcal{Y}| = m$. In this case we could set $\boldsymbol{s} = \boldsymbol{f}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}} \boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^m$ to be the mapping functional. We can calibrate the probabilities (through $\boldsymbol{s}$) to be "fair" in some way – e.g. that the probability of outputting various words denoting professions should be the same regardless of the gender of pronouns used in the prompt. We note that we do not always use the dependence of $\boldsymbol{s}$ on all of its inputs and assign different $\boldsymbol{s}$ in different settings.

We write $\mathcal{G}$ to denote the class of functions that encode demographic subgroups (along with other information) and for each $\boldsymbol{g} \in \mathcal{G}$, $\boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x}) \in \mathbb{R}^l$, consistent with the dimension of $\boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D})$ so that we can calibrate over every dimension of $\boldsymbol{s}$. For example, when $l = 1$, $\mathcal{G}$ can be set to be the indicator function of different sensitive subgroups of $\mathcal{X}$. Alternately, in fair text generation tasks, when the dimension of $\boldsymbol{s}$ equals the size of the set $\mathcal{Y}$, denoted as $l = m$, we can set the vector $\boldsymbol{g} \in \mathcal{G}$ to have a value of 1 in the dimensions corresponding to certain types of sensitive words, and 0 in all other dimensions.

We now formally introduce the $(\boldsymbol{s}, \mathcal{G}, \alpha)$-Generalized Multicalibration ($(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC) definition.

Definition 3.1 ($(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC)**.** Let $\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}$ denote the feature vector, the scoring function, the label vector, and the joint distribution of $(\boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y})$ respectively. Given a function class $\mathcal{G}$, mapping functional $\boldsymbol{s}$, and a threshold $\alpha > 0$, we say $\boldsymbol{f}$ satisfies $(\boldsymbol{s}, \mathcal{G}, \alpha)$-Generalized Multicalibration ($(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC) if

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle] \leq \alpha, \quad \forall \boldsymbol{g} \in \mathcal{G}.$$

$(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC is a flexible framework that can instantiate many existing multi-group fairness notions, including s-HappyMap (Deng et al., 2023), property multicalibration (Noarov & Roth, 2023), calibrated multivalid coverage (Jung et al., 2022) and outcome indistinguishability (Dwork et al., 2021). More generally, compared to these notions, $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC extends the literature in two ways. First, it allows the functions $\boldsymbol{s}$ and $\boldsymbol{g}$ to be multi-dimensional (most prior definitions look similar, but with 1-dimensional $\boldsymbol{s}$ and $\boldsymbol{g}$ functions). Second, the function $\boldsymbol{s}$ here is more general and allowed to be a *functional* of $\boldsymbol{f}$ (rather than just a function of $\boldsymbol{f}(\boldsymbol{x})$, the evaluation of $\boldsymbol{f}$ at $\boldsymbol{x}$). These generalizations will be important in our applications.

3.2. Algorithm and Convergence Results

To achieve $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC, we present the $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC Algorithm, which can be seen as a natural generalization of algorithms used for more specific notions of multicalibration in previous work (Úrsula Hébert-Johnson et al., 2018; Dwork et al., 2021; Jung et al., 2022; Deng et al., 2023):

Algorithm 1 $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC Algorithm

```
Input: step size $\eta > 0$, initialization $\boldsymbol{f}^{(0)} \in \mathcal{Q}$, max iteration $T$.
Initialization: $t = 0$.
while $t < T, \exists \boldsymbol{g}^{(t)} \in \mathcal{G}\ s.t:$ $\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\langle \boldsymbol{s}(\boldsymbol{f}^{(t)}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}^{(t)}(\boldsymbol{f}^{(t)}(\boldsymbol{x}), \boldsymbol{x})\rangle] > \alpha$ do
    Let $\boldsymbol{g}^{(t)} \in \mathcal{G}$ be an arbitrary function satisfying the condition in the while statement
    $\boldsymbol{f}^{(t+1)}(\boldsymbol{x}) = \mathrm{Proj}_{\mathcal{F}}\left(\boldsymbol{f}^{(t)}(\boldsymbol{x}) - \eta \boldsymbol{g}^{(t)}(\boldsymbol{f}^{(t)}(\boldsymbol{x}), \boldsymbol{x})\right)$
    $t = t + 1$
end while
Output: $\boldsymbol{f}^{(t)}$
```

It is worth noting that our goal involves functionals concerning our objective function $\boldsymbol{f}$ in order to capture its global properties. We aim to find a function $\boldsymbol{f}$ such that a functional associated with it (obtained by taking the expectation over $\boldsymbol{x}$) satisfies the inequalities we have set to meet different fairness demands. Before delving into the main part of our convergence analysis, we introduce some definitions related to functionals. Examples of these definitions can be found in the appendixB.

Definition 3.2 (The derivative of a functional)**.** Given a function $\boldsymbol{f} : \mathcal{X} \to \mathcal{F}$, consider a functional $\mathcal{L}(\boldsymbol{f}, \mathcal{D}) : \mathcal{Q} \times \mathcal{P} \to \mathbb{R}$, where $\mathcal{Q}$ is the function space of $\boldsymbol{f}$, $\mathcal{P}$ is a distribution space over $\mathcal{X}$. Assume that $\mathcal{L}$ follows the formulation that $\mathcal{L}(\boldsymbol{f}, \mathcal{D}) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[L(\boldsymbol{f}(\boldsymbol{x}))]$. The derivative function of $\mathcal{L}(\boldsymbol{f}, \mathcal{D})$ with respect to $\boldsymbol{f}$, denoted as $\nabla_{\boldsymbol{f}}\mathcal{L}(\boldsymbol{f}, \mathcal{D}) : \mathcal{X} \to \mathcal{F}$, exists if $\forall \boldsymbol{w} \in \mathcal{Q}, \boldsymbol{y} \in \mathbb{R}^m, \mathcal{D} \in \mathcal{P}, \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\langle \nabla_{\boldsymbol{f}}\mathcal{L}(\boldsymbol{f}, \mathcal{D}), \boldsymbol{w}\rangle]$ $= \frac{\partial}{\partial \epsilon}\ \mathcal{L}(\boldsymbol{f} + \epsilon \boldsymbol{w}, \mathcal{D})\big|_{\epsilon=0}$.

In the following, we define the definition of convexity and smoothness of a functional.

Definition 3.3 (Convexity of a functional)**.** Let $\mathcal{L}$ and $\boldsymbol{f}$ be defined as in Definition 3.2. A functional $\mathcal{L}$ is convex with respect to $\boldsymbol{f}$ if for any $\boldsymbol{f_1}, \boldsymbol{f_2} \in \mathcal{Q}, \mathcal{L}(\boldsymbol{f_1}, \mathcal{D}) - \mathcal{L}(\boldsymbol{f_2}, \mathcal{D}) \geq \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\langle \nabla_{\boldsymbol{f}}\mathcal{L}(\boldsymbol{f_2}, \mathcal{D}), \boldsymbol{f_1} - \boldsymbol{f_2}\rangle]$.

Definition 3.4 ($K_{\mathcal{L}}$-smoothness of a functional)**.** Let $\mathcal{L}$ and $\boldsymbol{f}$ be defined as in Definition 3.2. A functional $\mathcal{L}$ is $K_{\mathcal{L}}-$smooth if for any $\boldsymbol{f_1}, \boldsymbol{f_2} \in \mathcal{Q}$, $\mathcal{L}(\boldsymbol{f_1}, \mathcal{D}) - \mathcal{L}(\boldsymbol{f_2}, \mathcal{D}) \leq \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\langle \nabla\mathcal{L}(\boldsymbol{f_2}, \mathcal{D}), \boldsymbol{f_1} - \boldsymbol{f_2}\rangle] + \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\frac{K_{\mathcal{L}}}{2}\|\boldsymbol{f_1} - \boldsymbol{f_2}\|^2]$.

We will prove that this algorithm converges and outputs an $\boldsymbol{f}$ satisfying $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC whenever the following assumptions are satisfied.

Assumptions

(1). There exists a potential functional $\mathcal{L}(\boldsymbol{f}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D})$, such that $\nabla_{\boldsymbol{f}}\mathcal{L}(\boldsymbol{f}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D})(\boldsymbol{x}) = \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D})$, and $\mathcal{L}(\boldsymbol{f}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D})$ is $K_{\mathcal{L}}$-smooth with respect to $\boldsymbol{f}$ for any $\boldsymbol{x} \in \mathcal{X}$.

(2). Let $\boldsymbol{f}^*(\boldsymbol{x}) \triangleq \mathrm{Proj}_{\mathcal{F}}\boldsymbol{f}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. For any $\boldsymbol{f} \in \mathcal{Q}, \mathcal{L}(\boldsymbol{f}^*, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}) \leq \mathcal{L}(\boldsymbol{f}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D})$.

(3). There exists a positive number B, such that for all $\boldsymbol{g} \in \mathcal{G}$ and all $\boldsymbol{f} \in \mathcal{Q}$, $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\|\boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\|^2] \leq B$.

(4). There exists two numbers C_l, C_u such that for all $\boldsymbol{f} \in \mathcal{Q}, \quad \mathcal{L}(\boldsymbol{f}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}) \geq C_l, \mathcal{L}(\boldsymbol{f}^{(0)}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}) \leq C_u$.

Assumption (1) says that a potential functional $\mathcal{L}$ exists and it satisfies a $K_{\mathcal{L}}$-smoothness condition with respect to $\boldsymbol{f}$. For example, when $\boldsymbol{f}$ is a predicted distribution, we often set $\boldsymbol{s} = \boldsymbol{f}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\boldsymbol{f}(\boldsymbol{x})$. In this situation, $\mathcal{L} = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\frac{1}{2}\|\boldsymbol{f}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\boldsymbol{f}(\boldsymbol{x})\|^2]$ satisfies the assumption.

Assumption (2) states that the potential function decreases when projected with respect to $\boldsymbol{f}$. A specific example is when $\mathcal{F} = \mathcal{Y} = [0, 1]$ and $\mathcal{L} = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}|f(\boldsymbol{x}) - y|^2$.

Assumption (3) states that the ℓ_2-norm of the functions in $\mathcal{G}$ is uniformly bounded. It always holds when $\mathcal{G}$ contains indicator functions, which is the most common case in fairness-motivated problems (these are usually the indicator functions for subgroups of the data).

Assumption (4) says that the potential functional $\mathcal{L}$ is lower bounded and this generally holds true when $\mathcal{L}$ is convex. One concrete example is when $s(f(\boldsymbol{x}), h, y) = f(\boldsymbol{x}) - y$ and we have $\mathcal{L}(f, h, y, \mathcal{D}) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[(f(\boldsymbol{x}) - y)^2]$, which is lower bounded by 0.

Theorem 3.5. *Under Assumptions 1-4, the $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC Algorithm with a suitably chosen $\eta = \mathcal{O}(\alpha/(K_{\mathcal{L}}B))$ converges in $T = \mathcal{O}(\frac{2K_{\mathcal{L}}(C_u - C_l)B}{\alpha^2})$ iterations and outputs a function $\boldsymbol{f}$ satisfying*

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle] \leq \alpha, \forall \boldsymbol{g} \in \mathcal{G}.$$

The proof is provided in Appendix C. At a high level, if we consider $\boldsymbol{g}$ as a generalized direction vector and $\boldsymbol{s}$ as the gradient of $\mathcal{L}$, each violation can be interpreted as detecting a direction where the first-order difference of $\mathcal{L}$ is significant. By introducing the assumption of smoothness, our update can result in a decrease in $\mathcal{L}$ that exceeds a constant value. Since $\mathcal{L}$ is lower bounded by assumption, the updates can terminate as described.

3.3. Finite-Sample Results

We have presented Algorithm 1 as if we have direct access to the true data distribution $\mathcal{D}$. In practice, we only have a finite calibration set D, whose data is sampled $i.i.d$ from $\mathcal{D}$. In this subsection, we show how a variant of Algorithm 1 achieves the same goal from finite samples.

First, we introduce a useful measure which we call the *dimension of the function class*, as similarly defined in (Kim et al., 2019; Deng et al., 2023). For a dataset D, we use $\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim D}$ to denote the empirical expectation over D. We need T datasets in all and we assume that the whole sample size is m (m/T for each dataset).

Definition 3.6 (Dimension of the function class)**.** We use $d(\mathcal{G})$ to denote the dimension of class $\mathcal{G}$, defined to be a quantity such that if the sample size $m \geq C_1 \frac{d(\mathcal{G}) + \log(1/\delta)}{\alpha^2}$, then a random sample S_m of m elements from $\mathcal{D}$ guarantees uniform convergence over $\mathcal{G}$ with error at most α with failure probability at most δ. That is, for any fixed $\boldsymbol{f}$ and fixed $\boldsymbol{s}$ with $\|\boldsymbol{s}\|_\infty \leq C_2$ ($C_1, C_2 > 0$ are universal constants):

$$\begin{aligned}\sup_{\boldsymbol{g}\in\mathcal{G}} &|\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle] \\ &- \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim S_m}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle]| \leq \alpha.\end{aligned}$$

A discussion of this definition is given in the appendix.

We now give the finite sample version of the $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC Algorithm and its convergence results below. The detailed proof is in the appendix; we use the uniform convergence guarantee arising from Definition 3.6 to relate the problem to its distributional counterpart.

Algorithm 2 $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC Algorithm (Finite Sample)

```
Input: step size $\eta > 0$, initialization $\boldsymbol{f}^{(0)}(\boldsymbol{x}) \in \mathcal{F}$, validation datasets $D_{[2T]}$, max iteration $T$.
Initialization: $t = 0$.
while $t < T, \exists \boldsymbol{g}^{(t)} \in \mathcal{G}, s.t.:$ $\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim D_{2t-1}}[\langle \boldsymbol{s}(\boldsymbol{f}^{(t)}(\boldsymbol{x}), \boldsymbol{h}, \boldsymbol{y}, D_{2t}), \boldsymbol{g}^{(t)}(\boldsymbol{f}^{(t)}(\boldsymbol{x}), \boldsymbol{x})\rangle] > \frac{3}{4}\alpha$ do
    Let $\boldsymbol{g}^{(t)} \in \mathcal{G}$ be an arbitrary function satisfying the condition in the while statement
    $\boldsymbol{f}^{(t+1)}(\boldsymbol{x}) = \mathrm{Proj}_{\mathcal{F}}\left(\boldsymbol{f}^{(t)}(\boldsymbol{x}) - \eta\boldsymbol{g}^{(t)}(\boldsymbol{f}^{(t)}(\boldsymbol{x}), \boldsymbol{x})\right)$
    $t = t + 1$
end while
Output: $\boldsymbol{f}^{(t)}$
```

Theorem 3.7. *Under the assumptions 1-4 given in section 3, suppose we run Algorithm 2 with a suitably chosen $\eta = \mathcal{O}\left(\alpha/\left(\kappa_{\mathcal{L}}B\right)\right)$ and sample size $m = \mathcal{O}\left(T \cdot \frac{d(\mathcal{G}) + \log(T/\delta)}{\alpha^2}\right)$, then with probability at least $1 - \delta$, the algorithm converges in $T = \mathcal{O}\left(\left(C_u - C_l\right)\kappa_{\mathcal{L}}B/\alpha^2\right)$ steps and returns a function* f *satisfying:*

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle] \leq \alpha, \forall \boldsymbol{g} \in \mathcal{G}.$$

4. Applications

In this section, we explore three applications of our framework: De-biased text generation in language modeling – where the output function is multi-dimensional and can't be addressed in other frameworks, uncertainty quantification in hierarchical classification — in which we can offer prediction set conditional coverage guarantees, and group-wise false-positive rate control in image segmentation. We begin by outlining the challenges related to fairness and robustness inherent to these applications. Subsequently, we illustrate how to integrate these challenges within the $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC framework, enabling their resolution through Algorithm 1.

4.1. De-Biased Text Generation

This section applies our framework to fair word prediction in language modelling. We think of a language model as a function that maps prompts to a distribution over the next word. More specifically, we write $\boldsymbol{x} \in \mathcal{X}$ to denote a prompt, given which the language model outputs a distribution over the vocabulary, denoted by $\mathcal{Y}$. Namely, the language model generates the probability vector $\boldsymbol{h}(\boldsymbol{x}) \in \Delta\mathcal{Y}$, and then samples a word (output) following $o(\boldsymbol{x}) \sim \boldsymbol{h}(\boldsymbol{x})$. Previous studies (Lu et al., 2018; Hoffmann et al., 2022) demonstrated the pervasive presence of gender bias in contemporary language models. Our objective in this section is to mitigate this issue through an approach that post-processes $\boldsymbol{h}(\boldsymbol{x})$ to a probability distribution $\boldsymbol{p}(\boldsymbol{x}) \in \Delta\mathcal{Y}$ that has better fairness properties in specific ways. To take advantage of the information in initial language model, $\boldsymbol{p}$ is initialized at $\boldsymbol{h}$.

At the high level, we aim to produce $\boldsymbol{p}(\boldsymbol{x})$ so that the probabilities of certain groups of words remain the same whether the prompt includes male-indicating words or female-indicating words. For example, we might not want "He was a __" to be completed with "doctor" more frequently than "She was a __" to be completed with "doctor". We define an attribute set U as a collection of specific sensitive words and $\mathcal{U}$ to be the set of all U, which stands for different kinds of sensitive words. Following (Lu et al., 2018; Hoffmann et al., 2022), we measure the bias of the model on sensitive attribute U by $|\mathbb{P}(o(\boldsymbol{x}) \in U|\boldsymbol{x} \in F) - \mathbb{P}(o(\boldsymbol{x}) \in U|\boldsymbol{x} \in M)|$, where the probability is taken over $o(\boldsymbol{x}) \sim \boldsymbol{p}(\boldsymbol{x})$, and $\boldsymbol{x} \in F$ and $\boldsymbol{x} \in M$ denotes that $\boldsymbol{x}$ indicates female and male pronouns respectively.

Suppose the marginal distribution over prompt $\boldsymbol{x}$ (which is drawn uniformly from the given corpus) satisfies that $\mathbb{P}(\boldsymbol{x} \in F), \mathbb{P}(\boldsymbol{x} \in M) \geq \gamma$ for some positive constant $\gamma > 0$, we get:

$$\begin{aligned}&|\mathbb{P}(o(\boldsymbol{x}) \in U|\boldsymbol{x} \in F) - \mathbb{P}(o(\boldsymbol{x}) \in U|\boldsymbol{x} \in M)|\\ &\leq \frac{1}{\gamma}(|\mathbb{P}(\boldsymbol{x} \in F)(\mathbb{P}(o(\boldsymbol{x}) \in U|\boldsymbol{x} \in F) - \mathbb{P}(o(\boldsymbol{x}) \in U))|\\ &+ |\mathbb{P}(\boldsymbol{x} \in M)(\mathbb{P}(o(\boldsymbol{x}) \in U|\boldsymbol{x} \in M) - \mathbb{P}(o(\boldsymbol{x}) \in U))|). \quad (1)\end{aligned}$$

As a result, we only need to control the terms on the right side of (1) instead. More specifically, we want to calibrate the output so that for any subset $U \in \mathcal{U} \subset \mathcal{Y}$ (e.g., gender-stereotyped professions) and subgroups $A \in \mathcal{A} \subset \mathcal{X}$ (e.g., gender-related pronouns),

$$|\mathbb{P}(\boldsymbol{x} \in A) \cdot [\mathbb{P}(o(\boldsymbol{x}) \in U|\boldsymbol{x} \in A) - \mathbb{P}(o(\boldsymbol{x}) \in U)]| \leq \alpha.$$

To better understand this fairness notion, let us consider a toy example where $\mathcal{X} = \{\text{he, she, his, her}\}$, $\mathcal{A} = \{\{\text{he,his}\}, \{\text{she,her}\}\}$, $\mathcal{Y} = \{\text{lawyer, doctor, dream, nurse}\}$, $\mathcal{U} = \{\{\text{lawyer, doctor}\}, \{\text{nurse}\}\}$. Our aim is to calibrate the output so that $|\mathbb{P}[o(\boldsymbol{x}) \in \{\text{lawyer, doctor}\}|x \in \{\text{she, her}\}] - \mathbb{P}[o(\boldsymbol{x}) \in \{\text{lawyer, doctor}\}]| \leq \alpha$ and $|\mathbb{P}[o(\boldsymbol{x}) \in \{\text{lawyer, doctor}\}|x \in \{\text{he, his}\}] - \mathbb{P}[o(\boldsymbol{x}) \in \{\text{lawyer, doctor}\}]| \leq \alpha$. We can define $\mathcal{V} \triangleq \{(1, 1, 0, 0), (0, 0, 0, 1)\}$ to be the set of indicator vectors of sensitive attributes defined by $\mathcal{U}$.

Setting $\mathcal{G} \triangleq \{\mathbb{1}_{\{\boldsymbol{x} \in A\}}\boldsymbol{v} : A \in \mathcal{A}, \boldsymbol{v} \in \mathcal{V}\} \cup \{-\mathbb{1}_{\{\boldsymbol{x} \in A\}}\boldsymbol{v} : A \in \mathcal{A}, \boldsymbol{v} \in \mathcal{V}\}$, this problem can be cast in the GMC framework, and leads to the following theorem:

Theorem 4.1. *Assuming that $\boldsymbol{x}$ is a prompt that is uniformly drawn from the given corpus, and $\boldsymbol{h}$ is given by any fixed language model and the size of the largest attribute set in $\mathcal{U}$ is upper bounded by B. With a suitably chosen $\eta = \mathcal{O}(\alpha/B)$, our algorithm halts after $T = \mathcal{O}(B/\alpha^2)$ iterations and outputs a function $\boldsymbol{p}$ satisfying: $\forall A \in \mathcal{A}, U \in \mathcal{U}$, when $o(\boldsymbol{x}) \sim \boldsymbol{p}(\boldsymbol{x})$, $\sup_{A \in \mathcal{A}} |\mathbb{P}(\boldsymbol{x} \in A) \cdot [\mathbb{P}(o(\boldsymbol{x}) \in U|\boldsymbol{x} \in A) - \mathbb{P}(o(\boldsymbol{x}) \in U)]| \leq \alpha$.*

For the finite-sample counterpart, by applying theorem 3.7, the sample complexity required in this setting is $\mathcal{O}(\frac{\log(2|\mathcal{U}||\mathcal{A}|) + \log(\frac{1}{\delta})}{\alpha^2})$.

4.2. Prediction-Set Conditional Coverage in Hierarchical Classification

Hierarchical classification is a machine learning task where the labels are organized in a hierarchical tree structure (Tieppo et al., 2022). More specifically, at the most granular level, predictions are made using labels on the leaves of the tree. These leaves are grouped together into semantically meaningful categories through their parent nodes, which are, in turn, grouped together through their parents, and so on up to the root of the tree. Such a tree structure allows us—when there is uncertainty as to the correct label—to predict intermediate nodes, which correspond to predicting *sets* of labels — the set of leaves descended from the intermediate node — giving us a way to quantify the uncertainty of our predictions. Our goal is to produce such set-valued predictions that have a uniform coverage rate conditional on the prediction we make, where a prediction set is said to "cover" the true label if the true label is a descendent of (or equal to) the node we predicted.

For example, in a K-class hierarchical text classification problem, our input $\boldsymbol{x} \in \mathcal{X}$ is a document and the label is a leaf node y on a classification tree with nodes V and edges E. For simplicity, set $V = \{1, 2, ..., |V|\}$ where the first K indices $\{1, 2, .., K\}$ denote leaf nodes (so the groundtruth label $y \in \{1, ..., K\}$). The tree is of depth H. For a given single-class classification model $\boldsymbol{h} : \boldsymbol{x} \to [0, 1]^K$, let $u(\boldsymbol{x}) \triangleq \arg\max_k h_k(\boldsymbol{x})$ denote the candidate with the highest score over all leaf nodes according to $\boldsymbol{h}$. $u(\boldsymbol{x})$ here corresponds to the most natural point prediction we might make given $\boldsymbol{h}$.

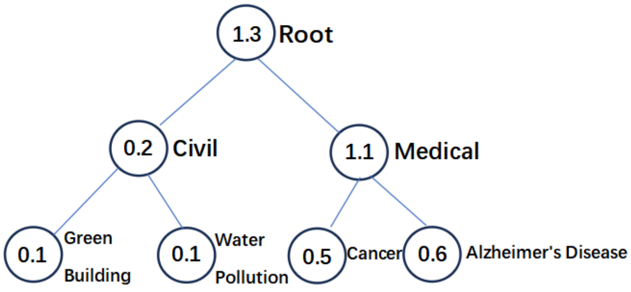


Figure 1. A demo of hierarchical text classification using a subset of labels from the *Web of Science* dataset. (Kowsari et al., 2017).

As a concrete example, in the tree diagram above, we map the set $\{1, 2, 3, 4, 5, 6, 7\}$ to represent the categories: Green Building, Water Pollution, Cancer, Alzheimer's Disease, Civil, Medical and Root. Consider a document $\boldsymbol{x}$ with the true label ‘Cancer’ and an initial model predicting scores $\boldsymbol{h}(\boldsymbol{x}) = (0.1, 0.1, 0.5, 0.6)$. If we used the scores to make a point prediction, we would be incorrect — the highest scoring label $u(\boldsymbol{x})$ is “Altzheimer’s disease”, and is wrong: $u(\boldsymbol{x}) \neq y$. If we output the parent node (‘Medical’) instead, our prediction would be less specific (a larger prediction set, here corresponding to both “Cancer” and “Alzheimer’s Disease”), but it would cover the true label. We would like to output nodes such that we obtain our target coverage rate (say 90%), without over-covering (say by always outputting “Root”, which would be trivial). Traditional conformal prediction methods (see (Angelopoulos & Bates, 2021) for a gentle introduction) give prediction sets that offer marginal guarantees of this sort, but not prediction-set conditional guarantees: i.e. they offer that for 90% of examples, we produce a prediction set that covers the true label. Recent applications of multicalibration related techniques ((Jung et al., 2021; Gupta et al., 2022; Bastani et al., 2022; Jung et al., 2022; Deng et al., 2023; Gibbs et al., 2023) are able to give “group conditional” coverage guarantees which offer (e.g.) 90% coverage as averaged over examples within each of a number of intersecting groups, but once again these methods are not able to offer prediction-set conditional guarantees. Prediction set conditional guarantees promise that for each prediction set that we produce, we cover 90% of example labels, *even conditional on the prediction set we offer*. This provides a stronger guarantee and precludes the possibility of our model being over-confident in some prediction sets and under-confident in others, as demonstrated in our experimental results.

We now define some useful functional notation. Let $\boldsymbol{A} : V \rightarrow V^H$ return the set of all the ancestor nodes of the input node. Let $q : V \times V \rightarrow V$ compute the nearest common ancestor of its two input nodes. Let $\boldsymbol{R} : \mathcal{X} \rightarrow \mathbb{R}^{|V|}$ be the function that computes for each node i, R_i, the sum of the raw scores $\boldsymbol{h}(\boldsymbol{x})$ assigned to each leaf that is a descendent of node i (or itself if i is a leaf). When needed, we may randomize $\boldsymbol{R}$ by letting $\boldsymbol{r}_i(\boldsymbol{x}) \triangleq \boldsymbol{R}_i(\boldsymbol{x}) + \epsilon_i(\boldsymbol{x})$, where $\epsilon(\boldsymbol{x})$ is an independent random variable with zero-mean and constant variance. We define a natural method to choose a node $o(\boldsymbol{x})$ to output given a scoring function $\boldsymbol{h}(\boldsymbol{x})$ and a threshold function $\lambda(\boldsymbol{x})$. We define $o(\boldsymbol{x}) \triangleq \arg\min_v \{d(v) : v \in \boldsymbol{A}(u(\boldsymbol{x})), \boldsymbol{r}_v < \lambda(\boldsymbol{x})\}$, where $d(v)$ denotes the depth of the node v in the tree. In other words, we output the highest ancestor i of $u(\boldsymbol{x})$ (which we recall is the point prediction we would make given $\boldsymbol{h}$ alone) whose cumulative score $\boldsymbol{r}_i$ is below some threshold — which we will select to obtain some target coverage probability. Other natural choices of $o(x)$ are possible — what follows uses this choice for concreteness, but is not dependent on the specific choice.

Recall that an output covers the label if it is the ancestor of the label or the label itself. Our goal is to find a $\lambda(\boldsymbol{x})$, such that the rate at which the output covers the label is roughly equal to a given target σ, not just overall, but conditional on the prediction set we output lying in various sets $\mathcal{U} \subset 2^V$:

$$|\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},y)\sim\mathcal{D}}[\mathbb{1}_{\{o(\boldsymbol{x})\in U\}}(\sigma - \mathbb{1}_{\{o(\boldsymbol{x}) \text{ covers } y\}})]| \leq \alpha, \forall U \in \mathcal{U}.$$

Back to our example, we can specify $\mathcal{U}$ in various ways. For example, we can set $\mathcal{U} = \{\{1,2,5\},\{3,4,6\}\}$ to require equal coverage cross the parent categories ‘Civil’ and ‘Medical’. Or, we can set $\mathcal{U} = \{\{1\},\{2\},\ldots,\{6\},\{7\}\}$ to obtain our target coverage rate σ conditionally on the prediction set we output for *all possible* prediction sets we might output.

We set $\mathcal{G} \triangleq \{\mathbb{1}_{\{o(\boldsymbol{x})\in U\}} : U \in \mathcal{U}\} \cup \{-\mathbb{1}_{\{o(\boldsymbol{x})\in U\}} : U \in \mathcal{U}\}$, fitting this problem into our GMC framework:

$$|\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},y)\sim\mathcal{D}}[g(o(\boldsymbol{x}))(\sigma - \mathbb{1}_{\{o(\boldsymbol{x}) \text{ covers } y\}})]| \leq \alpha, \forall g \in \mathcal{G}.$$

Using $\sum_{i=1}^{K} \mathbb{1}_{\{\boldsymbol{r}_{q(i,u)}(\boldsymbol{x})<\lambda\}}\mathbb{1}_{\{y=i\}} = \mathbb{1}_{\{o(\boldsymbol{x}) \text{ covers } y\}}$ and applying Algorithm 1, we obtain the following theorem:

Theorem 4.2. *Assume (1). $\forall u, \forall i \in V, f_{r_i|\boldsymbol{x}}(u) \leq K_p$, where $f_{r_i|\boldsymbol{x}}(u)$ denotes the density function of r_i conditioned on $\boldsymbol{x}$; (2). There exists a real number $M > 0$ such that $\forall i \in V, r_i \in [-M, M]$. With a suitably chosen $\eta = \mathcal{O}(\alpha/K_P)$, our algorithm halts after $T = \mathcal{O}(K_P M/\alpha^2)$ iterations and outputs a function λ satisfying that $\forall U \in \mathcal{U}$,*

$$|\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},y)\sim\mathcal{D}}[\mathbb{1}_{\{o(\boldsymbol{x})\in U\}}(\sigma - \mathbb{1}_{\{o(\boldsymbol{x}) \textit{ covers } y\}})]| \leq \alpha.$$

Applying theorem 3.7, we can see that the sample complexity for the finite-sample version of the algorithm is $\mathcal{O}(\frac{\log(2|\mathcal{U}|)+\log(\frac{1}{\delta})}{\alpha^2})$.

4.3. Fair FNR Control in Image Segmentation

In image segmentation, the input is an image of $m = w \times l$ (w for width and l for length) pixels and the task is to distinguish the pixels corresponding to certain components of the image, e.g., tumors in a medical image, eyes in the picture of a face, etc. As pointed out in (Lee et al., 2023), gender and racial biases are witnessed when evaluating image segmentation models. Among the common evaluations of image segmentation, we consider the False Negative Rate (FNR), defined as $\frac{\text{False Negatives}}{\text{False Negatives}+\text{True Positives}}$. In image segmentation when O, O' denotes the set of the actual selected segments and the predicted segments respectively, FNR $= 1 - \frac{|O\cap O'|}{|O|}$.

We write $\boldsymbol{x} \in \mathcal{X}$ to denote the input, which includes both image and demographic group information and $\boldsymbol{y} \in \{0,1\}^m$ to denote the label, which is a binary vector denoting the true inclusion of each of the m pixels. To yield the prediction of $\boldsymbol{y}$, namely $\hat{\boldsymbol{y}} \in \{0,1\}^m$, a scoring function $\boldsymbol{h}(\boldsymbol{x}) \in \mathbb{R}^m$ and a threshold function $\lambda(\boldsymbol{x})$ are needed, so that $\hat{y}_i = \mathbb{1}_{\{h_i(\boldsymbol{x})>\lambda(\boldsymbol{x})\}}$ for $i \in [m]$. As in Section 4.2, for technical reasons we may randomize h_i by perturbing it with a zero-mean random variable of modest scale. Our objective is to determine the threshold function $\lambda(\boldsymbol{x})$.

In the context of algorithmic fairness in image segmentation, one specific application is face segmentation, where the objective is to precisely identify and segment regions containing human faces within an image. The aim is to achieve accurate face segmentation while ensuring consistent levels of precision across various demographic groups defined by sensitive attributes, like gender and race. Thus, our objective is to determine the function $\lambda(\boldsymbol{x})$ that ensures multi-group fairness in terms of the FNR — a natural multi-group fairness extension of the FNR control problem for image segmentation studied in (Angelopoulos et al., 2023).

Letting $\mathcal{A}$ be the set of sensitive subgroups of $\mathcal{X}$, our goal is to ensure that the FNR across different groups are approximately $(1-\sigma)$ for some prespecified $\sigma > 0$:

$$|\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\mathbb{1}_{\{\boldsymbol{x}\in A\}}(1-\tfrac{|O\cap O'|}{|O|}-\sigma)]| \leq \alpha, \quad \forall A \in \mathcal{A}.$$

We can write $|O \cap O'| = \sum_{i=1}^m y_i \mathbb{1}_{\{h_i(\boldsymbol{x})>\lambda(\boldsymbol{x})\}}$, so the object is converted to

$$\sup_{A\in\mathcal{A}} |\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\mathbb{1}_{\{x\in A\}}(1-\frac{\sum_{i=1}^m y_i \mathbb{1}_{\{h_i(\boldsymbol{x})>\lambda(\boldsymbol{x})\}}}{\sum_{i=1}^m y_i}-\sigma)]| \leq \alpha.$$

Let $s(\lambda,\boldsymbol{x},\boldsymbol{h},\boldsymbol{y}) = 1 - \frac{\sum_{i=1}^m y_i \mathbb{1}_{\{h_i(\boldsymbol{x})>\lambda(\boldsymbol{x})\}}}{\sum_{i=1}^m y_i} - \sigma$ and $\mathcal{G} \triangleq \{\pm\mathbb{1}_{\{\boldsymbol{x}\in A\}} : A \in \mathcal{A}\}$. Rewriting the inequality we get: $\sup_{g\in\mathcal{G}} \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[g(\lambda(\boldsymbol{x}),\boldsymbol{x})s(\lambda,\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})] \leq \alpha$. Cast in the GMC framework, we obtain the following result:

Theorem 4.3. *Assume (1) For all $i \in [n]$, $|h_i| \leq M$ for some universal constant $M > 0$; (2) the density function of h_i conditioned on x is upper bounded by some universal constant $K_p > 0$. Let C be the set of sensitive subgroups of $\mathcal{X}$. Then with a suitably chosen $\eta = \mathcal{O}(\alpha/(K_P))$, the algorithm halts after $T = \mathcal{O}(\frac{2K_P M}{\alpha})$ iterations and outputs a function λ satisfying:*

$$|\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\mathbb{1}_{\{\boldsymbol{x}\in A\}}(1-\tfrac{|O\cap O'|}{|O|}-\sigma)]| \leq \alpha, \quad \forall A \in \mathcal{A}.$$

Similar to the previous two applications, by applying Theorem 3.7 for the finite-sample version of the algorithm, the sample complexity required is $\mathcal{O}(\frac{\log(2|\mathcal{A}|)+\log(\frac{1}{\delta})}{\alpha^2})$.

We note that equalizing false negative rates across groups can be achieved trivially by setting λ to be large enough so that the FNR is equalized (at 0) — which would of course destroy the accuracy of the method. Thus when we set an objective like this, it is important to empirically show that not only does the method lead to low disparity across false negative rates, but does so without loss in accuracy. The experiments that we carry out in Section 5 indeed bear this out.

5. Experiments

In this section, we conduct numerical experiments and evaluate the performance of our algorithms within each application from both the fairness and accuracy perspectives. We compare the results with baseline methods to assess their effectiveness. The code can be found in the supplementary material. For more detailed experiment settings and additional results, please refer to Appendix D. The code and data are stored in the repository `https://github.com/MisDrifter/GMC`.

5.1. De-Biased Text Generation

In text generation, we consider two datasets and run experiments separately. The first dataset is the corpus data from Liang et al. (2021), which extracts sentences with both terms indicative of biases (e.g., gender indicator words) and attributes (e.g., professions) from real-world articles. The second dataset is made up of synthetic templates based on combining words indicative of bias targets and attributes with simple placeholder templates, e.g., “The woman worked as ...”; “The man was known for ...”, constructed in (Lu et al., 2019).

Then, we define two kinds of terms indicative of bias targets: female-indicator words and male-indicator words; we also define six types of attributes: female-adj words, male-adj words, male-stereotyped jobs, female-stereotyped jobs, pleasant words, and unpleasant words, by drawing on existing word lists in the fair text generation context (Caliskan et al., 2017) (Gonen & Goldberg, 2019).

Each input $\boldsymbol{x}$ is a sentence where sensitive attributes are masked. We use the BERT model (Devlin et al., 2018) to generate the initial probability distribution over the entire vocabulary for the word at the masked position, denoted by $\boldsymbol{h}(\boldsymbol{x})$. We then use our algorithm to post-process $\boldsymbol{h}(\boldsymbol{x})$ and obtain the function $\boldsymbol{p}(\boldsymbol{x})$, which is the calibrated probability of the output. We define two sets of prompts: A_{female} and A_{male} be the set of prompts containing female-indicator and male-indicator words, respectively. We aim to control the gender disparity gap $|\mathbb{P}(\boldsymbol{x} \in A) \cdot [\mathbb{P}(o(\boldsymbol{x}) \in U | \boldsymbol{x} \in A) - \mathbb{P}(o(\boldsymbol{x}) \in U)]|$ for $A \in \{A_{\text{female}}, A_{\text{male}}\}$.

Figure 2 plots the disparity gap for $A = A_{male}$ (the result for $A = A_{female}$ is deferred to the appendix due to space constraints). It is evident that our post-processing technique effectively limits the disparity between the probabilities of outputting biased terms related to different gender groups, ensuring that it remains consistently below a specified threshold value of $\alpha = 0.002$ (we will further discuss the way of choosing α in the Appendix D). Additionally, we assess the cross-entropy loss between the calibrated output distribution and the corresponding labels. Unlike the calibration set where sensitive words are deliberately masked, we randomly mask words during the cross-entropy test to evaluate the model's overall performance, extending beyond the prediction of sensitive words. The cross-entropy of the test set is 9.9291 before post-processing and 9.9285 after it, indicating that our algorithm does not reduce the accuracy of the model while reducing gender disparities. We would like to note that our algorithm is not designed to enhance accuracy but fairness while ensuring that the performance of cross-entropy does not deteriorate too much.

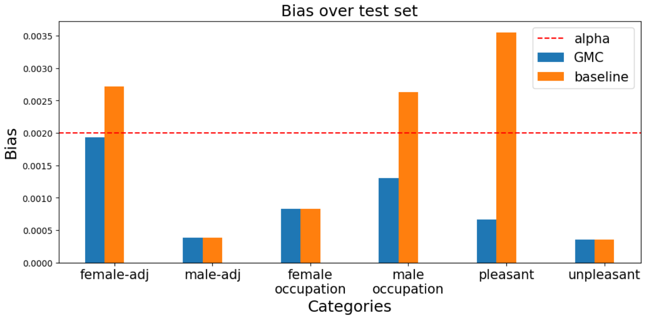


Figure 2. The bias on outputting different types of sensitive attributes measured on the corpus data. The results for the synthetic data are deferred to the appendix.

5.2. Prediction-Set Conditional Coverage in Hierarchical Classification

For hierarchical classification, we use the *Web of Science* dataset (Kowsari et al., 2017) that contains $46,985$ documents with 134 categories including 7 parent categories. We choose HiAGM (Wang et al., 2022) as the network to generate the initial scoring. Our algorithm is then applied to find the threshold function that yields a fair output.

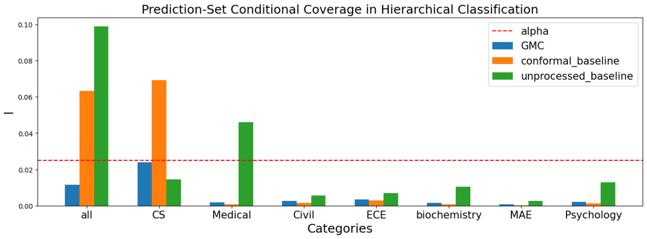


Figure 3. The deviation of prediction-set conditional coverage from the target.

We set our coverage target to be $\sigma = 0.95$ with a tolerance for coverage deviations of $\alpha = 0.025$. Equivalently put, our goal is that for each of the predictions, we aim to cover the true label with probability $95 \pm 2.5\%$, even *conditional on the prediction we make*. We choose naively outputting the leaf node (denoted as "unprocessed" in the figure) as one baseline and the split conformal method (Angelopoulos et al., 2023) as another baseline. Figure 3 shows that our method achieves coverage within the target tolerance for all predictions, while the two baselines fail to satisfy the coverage guarantee for predicting 'CS' and 'Medical'.

5.3. Fair FNR Control in Image Segmentation

We use the FASSEG (Khan et al., 2015) dataset and adopt the U-net (Ronneberger et al., 2015) network to generate the initial scoring function for each pixel, representing the predicted probability of this pixel corresponding to the signal. The dataset contains 118 human facial images and their semantic segmentations. We set our target FNR to be $\sigma = 0.075$ with a tolerance for deviations of $\alpha = 0.005$ and calibrate the FNR across different gender subgroups and racial subgroups. In addition, we compare with the method proposed in (Angelopoulos et al., 2023) that controls on-average FNR in a finite-sample manner based on the split conformal prediction method. The results yielded by U-net and the split conformal are plotted as baselines for comparison in Figure 4. Our algorithm demonstrates its effectiveness as the deviations of the FNRs of GMC from the target α across all subgroups are controlled below σ, while the baselines are found to perform poorly on male and white subgroups. Since equalizing FNR does not necessarily imply accuracy, we compute the accuracy of our model's output together with that of the baseline. The accuracy of our model, measured as the ratio of correctly predicted pixels to the total number of pixels, is 0.86. In comparison, the baseline models achieve an accuracy of 0.84 and 0.92, respectively. This result suggests that our algorithm empirically yields significant gains in mitigating FNR disparities without a significant sacrifice in accuracy.

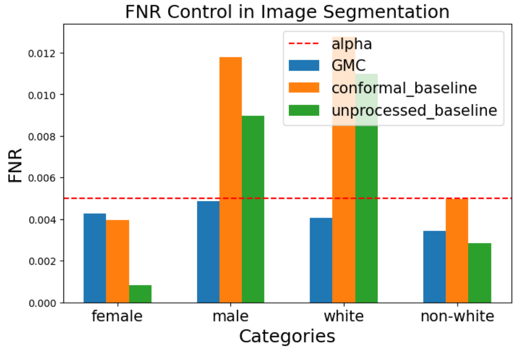


Figure 4. The deviation of the false negative rate from the target in image segmentation.

6. Conclusion and Discussion

This paper introduces the $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC framework for post-processing machine learning models (with multi-dimensional output) so that their predictions satisfy fairness guarantees across possibly intersecting groups. We apply this framework to three settings: equalizing false negative rates across demographic groups in image segmentation tasks, ensuring prediction-set conditional coverage in hierarchical classification, and post-processing language models to mitigate biases associated with protected contexts (e.g. gender-indicating pronouns) and stereotyped terms (e.g. gender-stereotyped professions). It would be an interesting direction to further apply our general framework to more fairness and robustness-motivated problems.

Impact Statement

This paper presents new, flexible, algorithms for obtaining a wide variety of technical notions of algorithmic fairness. Of course technical interventions of the sort we propose here are inherently limited, and cannot correct for problems that arise in gathering training data, choosing the problem to solve, or inappropriately deploying automated decision-making technologies. The kinds of techniques we develop here should be used with an understanding of their abilities and limitations.

References

Angelopoulos, A. N. and Bates, S. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., and Schuster, T. Conformal risk control, 2023.

Bastani, O., Gupta, V., Jung, C., Noarov, G., Ramalingam, R., and Roth, A. Practical adversarial multivalid conformal prediction. *Advances in Neural Information Processing Systems*, 35:29362–29373, 2022.

Bauschke, H. H. and Combettes, P. L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer, 2017.

Blum, A. and Lykouris, T. Advancing subgroup fairness via sleeping experts. *arXiv preprint arXiv:1909.08375*, 2019.

Caliskan, A., Bryson, J. J., and Narayanan, A. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017. doi: 10.1126/science.aal4230.

Chouldechova, A. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017.

Dawid, A. P. Calibration-based empirical probability. *The Annals of Statistics*, 13(4):1251–1274, 1985.

Deng, Z., Dwork, C., and Zhang, L. Happymap: A generalized multi-calibration method, 2023.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

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.

Dwork, C., Lee, D., Lin, H., and Tankala, P. From pseudorandomness to multi-group fairness and back, 2023.

Foster, D. P. and Hart, S. Smooth calibration, leaky forecasts, finite recall, and nash dynamics. *Games and Economic Behavior*, 109:271–293, 2018.

Foster, D. P. and Kakade, S. M. Calibration via regression. In *2006 IEEE Information Theory Workshop-ITW'06 Punta del Este*, pp. 82–86. IEEE, 2006.

Gibbs, I., Cherian, J. J., and Candès, E. J. Conformal prediction with conditional guarantees, 2023.

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.

Gonen, H. and Goldberg, Y. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 609–614, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1061. URL https://aclanthology.org/N19-1061.

Gopalan, P., Kim, M. P., Singhal, M. A., and Zhao, S. Low-degree multicalibration. In *Conference on Learning Theory*, pp. 3193–3234. PMLR, 2022.

Gupta, V., Jung, C., Noarov, G., Pai, M. M., and Roth, A. Online multivalid learning: Means, moments, and prediction intervals. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Dagstuhl Publishing, 2022.

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

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Jung, C., Lee, C., Pai, M., Roth, A., and Vohra, R. Moment multicalibration for uncertainty estimation. In *Conference on Learning Theory*, pp. 2634–2678. PMLR, 2021.

Jung, C., Noarov, G., Ramalingam, R., and Roth, A. Batch multivalid conformal prediction, 2022.

Kearns, M., Neel, S., Roth, A., and Wu, Z. S. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International conference on machine learning*, pp. 2564–2572. PMLR, 2018.

Kearns, M., Neel, S., Roth, A., and Wu, Z. S. An empirical study of rich subgroup fairness for machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*, pp. 100–109, 2019.

Khan, K., Mauro, M., and Leonardi, R. Multi-class semantic segmentation of faces. In *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 827–831, 2015. doi: 10.1109/ICIP.2015.7350915.

Kim, M. P., Ghorbani, A., and Zou, J. Multiaccuracy: Blackbox post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 247–254, 2019.

Kleinberg, J., Mullainathan, S., and Raghavan, M. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016.

Kowsari, K., Brown, D. E., Heidarysafa, M., Jafari Meimandi, K., , Gerber, M. S., and Barnes, L. E. Hdltex: Hierarchical deep learning for text classification. In *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*. IEEE, 2017.

Lee, T., Puyol-Antón, E., Ruijsink, B., Aitcheson, K., Shi, M., and King, A. P. An investigation into the impact of deep learning model choice on sex and race bias in cardiac mr segmentation. *arXiv preprint arXiv:2308.13415*, 2023.

Liang, P. P., Wu, C., Morency, L., and Salakhutdinov, R. Towards understanding and mitigating social biases in language models. *CoRR*, abs/2106.13219, 2021. URL https://arxiv.org/abs/2106.13219.

Lu, K., Mardziel, P., Wu, F., Amancharla, P., and Datta, A. Gender bias in neural natural language processing. *CoRR*, abs/1807.11714, 2018. URL http://arxiv.org/abs/1807.11714.

Lu, K., Mardziel, P., Wu, F., Amancharla, P., and Datta, A. Gender bias in neural natural language processing, 2019.

Noarov, G. and Roth, A. The statistical scope of multicalibration. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 26283–26310. PMLR, 2023. URL https://proceedings.mlr.press/v202/noarov23a.html.

Noarov, G., Ramalingam, R., Roth, A., and Xie, S. High-dimensional prediction for sequential decision making. *arXiv preprint arXiv:2310.17651*, 2023.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

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

Sandroni, A., Smorodinsky, R., and Vohra, R. V. Calibration with many checking rules. *Mathematics of operations Research*, 28(1):141–153, 2003.

Tieppo, E., Santos, R. R. d., Barddal, J. P., et al. Hierarchical classification of data streams: a systematic literature review. *Artificial Intelligence Review*, 55:3243–3282, 2022. doi: 10.1007/s10462-021-10087-z.

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.

Wang, Z., Wang, P., Liu, T., Lin, B., Cao, Y., Sui, Z., and Wang, H. Hpt: Hierarchy-aware prompt tuning for hierarchical text classification, 2022.

Zhao, S., Kim, M., Sahoo, R., Ma, T., and Ermon, S. Calibrating predictions to decisions: A novel approach to multi-class calibration. *Advances in Neural Information Processing Systems*, 34:22313–22324, 2021.

Úrsula Hébert-Johnson, Kim, M. P., Reingold, O., and Rothblum, G. N. Calibration for the (computationally-identifiable) masses, 2018.

A. Detailed Discussion of the dimension of the function class

We first state the concentration bounds to use in this section.

Theorem A.6 (Generalized Chernoff Bound)**.** *Let $\{X_i : i \in [n]\}$ be independent random variables satisfying $X_i \in [a_i, b_i]$. Let $X = \sum_{j=1}^n X_j, \mu = \mathbb{E}[X]$, then for $\lambda > 0$,*

$$\mathbb{P}(|X - \mu| \le \lambda) \le 2exp(-\frac{2\lambda^2}{\sum_{j=1}^n (b_i - a_i)^2}).$$

In this section, we will give the specific form of $d(\mathcal{G})$ for $\mathcal{G}$ used in some of our applications. In high-level, the Chernoff Bound is the basis for the derivation of $d(\mathcal{G})$.

We first restate Definition 1 here:

Definition A.7 (Dimension of the function class)**.** We use $d(\mathcal{G})$ to denote the dimension of an agnostically learnable class $\mathcal{G}$, such that if the sample size $m \ge C_1 \frac{d(\mathcal{G}) + \log(1/\delta)}{\alpha^2}$ for some universal constant $C_1 > 0$, then two independent random samples S_{m1} and S_{m2} from $\mathcal{D}$ guarantee uniform convergence over $\mathcal{G}$ with error at most α with failure probability at most δ, that is, for any fixed f and fixed s with $\|s\|_\infty \le C_2$ for some universal constant $C_2 > 0$:

$$\sup_{\boldsymbol{g} \in \mathcal{G}} \left| \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y}) \sim \mathcal{D}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle] - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y}) \sim S_{m1}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, S_{m2}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle] \right| \le \alpha.$$

The quantity $d(\mathcal{G})$ can be upper bounded by the VC dimension for boolean functions, and the metric entropy for real-valued functions.

When $\mathcal{G}$ is a finite function class (which is the case in our applications), we can establish the relation between $d(\mathcal{G})$ and $|\mathcal{G}|$ by the following theorem:

Theorem A.8. *When $\mathcal{G}$ is a finite function class, and for any $\boldsymbol{g} \in \mathcal{G}$, there exists a real number $A > 0$ such that $\|\boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\| \le A$ holds for any $\boldsymbol{f}(\boldsymbol{x}) \in \mathcal{F}$ and $\boldsymbol{x} \in \mathcal{X}$. We have $d(\mathcal{G}) = 2|\mathcal{G}|, C_1 = 2(AC_2)^2$.*

Proof. From the assumption, we know that

$$\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle \in [-AC_2, AC_2], \forall \|s\|_\infty \le C_2, \boldsymbol{g} \in \mathcal{G}, \boldsymbol{x} \in \mathcal{X}, \boldsymbol{f} \in \mathcal{Q}.$$

For any fixed $g \in \mathcal{G}$, apply theorem A.6, we have

$$m|(\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y}) \sim \mathcal{D}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle] - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y}) \sim S_{m1}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, S_{m2}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle]| \le \lambda.$$

with the probability of failure less than $2exp(-\frac{2\lambda^2}{\sum_{j=1}^m (AC_2 + AC_2)^2}) = 2exp(-\frac{2\lambda^2}{4m(AC_2)^2})$. Set $\lambda = \alpha m, m = \frac{2(AC_2)^2}{\alpha^2} \log(\frac{2|\mathcal{G}|}{\delta})$, We have

$$|\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y}) \sim \mathcal{D}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle] - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y}) \sim S_{m1}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, S_{m2}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle]| \le \alpha.$$

with the probability of failure less than $\frac{\delta}{|\mathcal{G}|}$.

Taking a union bound,

$$\sup_{\boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x}) \in \mathcal{G}} \left| \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y}) \sim \mathcal{D}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle] - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y}) \sim S_{m1}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, S_{m2}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle] \right| \le \alpha.$$

with the probability of failure less than δ.

So here we can set $d(\mathcal{G}) = 2|\mathcal{G}|, C_1 = 2(AC_2)^2$. □

B. Examples of the functional definitions

We recall the definitions given in the paper and give some examples for them to give more insights.

Definition B.1 (The derivative of a functional)**.** Given a function $\boldsymbol{f}:\mathcal{X}\to\mathcal{F}$, consider a functional $\mathcal{L}(\boldsymbol{f},\mathcal{D}):\mathcal{Q}\times\mathcal{P}\to\mathbb{R}$, where $\mathcal{Q}$ is the function space of $\boldsymbol{f}$, $\mathcal{P}$ is a distribution space over $\mathcal{X}$. Assume that $\mathcal{L}$ follows the formulation that $\mathcal{L}=\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[L(\boldsymbol{f}(\boldsymbol{x}))]$. The derivative function of $\mathcal{L}$ with respect to $\boldsymbol{f}$, denoted as $\nabla_{\boldsymbol{f}}\mathcal{L}(\boldsymbol{f},\mathcal{D}):\mathcal{X}\to\mathcal{F}$, exists if $\forall\boldsymbol{w}\in\mathcal{Q},\boldsymbol{y}\in\mathbb{R}^m,\mathcal{D}\in\mathcal{P},\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\langle\nabla_{\boldsymbol{f}}\mathcal{L}(\boldsymbol{f},\mathcal{D}),\boldsymbol{w}\rangle]$
$=\frac{\partial}{\partial\epsilon}\,\mathcal{L}(\boldsymbol{f}+\epsilon\boldsymbol{w},\mathcal{D})\big|_{\epsilon=0}$.

And it's defined to be a function satisfying the equation.

Example 1. *When $\mathcal{L}(f,\mathcal{D})=\mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}[\frac{1}{2}[f(\boldsymbol{x})-y]^2]$ (here y and $f(\boldsymbol{x})$ are 1-dimensional), $\nabla_f\mathcal{L}(f,\mathcal{D})(\boldsymbol{x})=f(\boldsymbol{x})-y$.*

Example 2. *When $\mathcal{L}(\boldsymbol{f},\mathcal{D})=\mathbb{E}_{\boldsymbol{x}}\frac{1}{2}\|\boldsymbol{f}(\boldsymbol{x})-\mathbb{E}_{\boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x})]\|^2$ (here $\boldsymbol{f}$ is multi-dimensional), $\nabla_{\boldsymbol{f}}\mathcal{L}(\boldsymbol{f},\mathcal{D})=\boldsymbol{f}(\boldsymbol{x})-\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\boldsymbol{f}(\boldsymbol{x})]$.*

Definition B.2 (Convexity of a functional)**.** Let $\mathcal{L}$ and $\boldsymbol{f}$ be defined as in Definition 3.2. A functional $\mathcal{L}$ is convex with respect to $\boldsymbol{f}$ if for any $\boldsymbol{f_1},\boldsymbol{f_2}\in\mathcal{Q},\mathcal{L}(\boldsymbol{f_1},\mathcal{D})-\mathcal{L}(\boldsymbol{f_2},\mathcal{D})\geq\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\langle\nabla_{\boldsymbol{f}}\mathcal{L}(\boldsymbol{f_2},\mathcal{D}),\boldsymbol{f_1}-\boldsymbol{f_2}\rangle]$.

Definition B.3 ($K_{\mathcal{L}}$-smoothness of a functional)**.** Let $\mathcal{L}$ and $\boldsymbol{f}$ be defined as in Definition 3.2. A functional $\mathcal{L}$ is $K_{\mathcal{L}}-$smooth if for any $\boldsymbol{f_1},\boldsymbol{f_2}\in\mathcal{Q}$, $\mathcal{L}(\boldsymbol{f_1},\mathcal{D})-\mathcal{L}(\boldsymbol{f_2},\mathcal{D})\leq\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\langle\nabla\mathcal{L}(\boldsymbol{f_2},\mathcal{D}),\boldsymbol{f_1}-\boldsymbol{f_2}\rangle]+\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\frac{K_{\mathcal{L}}}{2}\|\boldsymbol{f_1}-\boldsymbol{f_2}\|^2]$.

Example 3. *$\mathcal{L}(f,\mathcal{D})=\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\frac{1}{2}[f(\boldsymbol{x})-y]^2$ is 1-smooth and convex with respect to f.*

Example 4. *$\mathcal{L}(\boldsymbol{f},y)=\mathbb{E}_{\boldsymbol{x}}\frac{1}{2}\|\boldsymbol{f}(\boldsymbol{x})-\mathbb{E}_{\boldsymbol{x}}[\boldsymbol{f}(\boldsymbol{x})]\|^2$ is 1-smooth and convex with respect to $\boldsymbol{f}$.*

Example 5. *$\mathcal{F}=\Delta Y$ and $\mathcal{L}(\boldsymbol{f},\boldsymbol{h},y)=\frac{1}{2}\|\boldsymbol{f}-\mathbb{E}_{\boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x})\|^2$. Then $\boldsymbol{f}^*=\mathbb{E}_{\boldsymbol{x}}\boldsymbol{f}(\boldsymbol{x})\in\mathcal{F}$ and assumption (2) is satisfied in this situation.*

C. Proof of the theorems

Assumptions

1. There exists a potential functional $\mathcal{L}(\boldsymbol{f},\boldsymbol{h},\boldsymbol{y},\mathcal{D})$, such that $\nabla_{\boldsymbol{f}}\mathcal{L}(\boldsymbol{f},\boldsymbol{h},\boldsymbol{y},\mathcal{D})(\boldsymbol{x})=\boldsymbol{s}(\boldsymbol{f},\boldsymbol{x},\boldsymbol{h},\boldsymbol{y},\mathcal{D})$, and $\mathcal{L}(\boldsymbol{f},\boldsymbol{h},\boldsymbol{y},\mathcal{D})$ is $K_{\mathcal{L}}$-smooth with respect to $\boldsymbol{f}$ for any $\boldsymbol{x}\in\mathcal{X}$.
2. Let $\boldsymbol{f}^*(\boldsymbol{x})\triangleq\mathrm{Proj}_{\mathcal{F}}\boldsymbol{f}(\boldsymbol{x})$ for all $\boldsymbol{x}\in\mathcal{X}$. For any $\boldsymbol{f}\in\mathcal{Q}$, $\mathcal{L}(\boldsymbol{f}^*,\boldsymbol{h},\boldsymbol{y},\mathcal{D})\leq\mathcal{L}(\boldsymbol{f},\boldsymbol{h},\boldsymbol{y},\mathcal{D})$.
3. There exists a positive number B, such that for all $\boldsymbol{g}\in\mathcal{G}$ and all $\boldsymbol{f}\in\mathcal{Q}$, $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\|\boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{x})\|^2]\leq B$.
4. There exists two numbers C_l,C_u such that for all $\boldsymbol{f}\in\mathcal{Q}$, $\quad\mathcal{L}(\boldsymbol{f},\boldsymbol{h},\boldsymbol{y},\mathcal{D})\geq C_l$, $\mathcal{L}(\boldsymbol{f}^{(0)},\boldsymbol{h},\boldsymbol{y},\mathcal{D})\leq C_u$.

Theorem C.1. *Under the assumptions above, the $(\boldsymbol{s},\mathcal{G},\alpha)$-GMC Algorithm with a suitably chosen $\eta=\mathcal{O}(\alpha/(K_{\mathcal{L}}B))$ converges in $T=\mathcal{O}(\frac{2K_{\mathcal{L}}(C_u-C_l)B}{\alpha^2})$ iterations and outputs a function $\boldsymbol{f}$ satisfying*

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\langle\boldsymbol{s}(\boldsymbol{f},\boldsymbol{x},\boldsymbol{h},\boldsymbol{y},\mathcal{D}),\boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{x})\rangle]\leq\alpha,\forall\boldsymbol{g}\in\mathcal{G}.$$

Proof. According to the selection of $\boldsymbol{g}^{(t)}$, we have

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\langle\boldsymbol{s}(\boldsymbol{f}^{(t)},\boldsymbol{x},\boldsymbol{h},\boldsymbol{y},\mathcal{D}),\boldsymbol{g}^{(t)}(\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{x})\rangle]>\alpha.$$

$$\begin{aligned}
\mathcal{L}(\boldsymbol{f}^{(t)},\boldsymbol{h},\boldsymbol{y},\mathcal{D})-\mathcal{L}(\boldsymbol{f}^{(t+1)},\boldsymbol{x},h,\boldsymbol{y},\mathcal{D})\geq&\mathcal{L}(\boldsymbol{f}^{(t)},\boldsymbol{h},\boldsymbol{y},\mathcal{D})-\mathcal{L}(\boldsymbol{f}^{(t)}-\eta\boldsymbol{g}^{(t)},\boldsymbol{h},\boldsymbol{y},\mathcal{D})\quad\text{(assumption 2)}\\
\geq&\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\langle\nabla_{\boldsymbol{f}}\mathcal{L}(\boldsymbol{f}^{(t)},\boldsymbol{h},\boldsymbol{y},\mathcal{D})(\boldsymbol{x}),\eta\boldsymbol{g}^{(t)}(\boldsymbol{f}^{(t)}(\boldsymbol{x}),\boldsymbol{x})\rangle]\\
&-\frac{\eta^2K_{\mathcal{L}}}{2}\mathbb{E}_{\boldsymbol{x}}[\|\boldsymbol{g}(\boldsymbol{f}^{(t)}(\boldsymbol{x}),\boldsymbol{x})\|^2]\quad\text{(assumption 1)}\\
=&\eta\mathbb{E}_{\boldsymbol{x},\boldsymbol{h},\boldsymbol{y}\sim\mathcal{D}}[\langle\boldsymbol{s}(\boldsymbol{f}^{(t)},\boldsymbol{x},\boldsymbol{h},\boldsymbol{y}),\boldsymbol{g}^{(t)}(\boldsymbol{f}(\boldsymbol{x}),\boldsymbol{x})\rangle]-\frac{\eta^2K_{\mathcal{L}}}{2}\mathbb{E}_{\boldsymbol{x}}[\|\boldsymbol{g}(\boldsymbol{f}^{(t)}(\boldsymbol{x}),\boldsymbol{x})\|^2]\\
\geq&\eta\alpha-\frac{\eta^2K_{\mathcal{L}}}{2}\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim D}[\|\boldsymbol{g}(\boldsymbol{f}^{(t)}(\boldsymbol{x}),\boldsymbol{x})\|^2].
\end{aligned}$$

Set $\eta=\alpha/(K_{\mathcal{L}}B)$ and use assumption 3, we get

$$\mathcal{L}(\boldsymbol{f}^{(t)},\boldsymbol{h},\boldsymbol{y},\mathcal{D})-\mathcal{L}(\boldsymbol{f}^{(t+1)},\boldsymbol{h},\boldsymbol{y},\mathcal{D})\geq\frac{\alpha^2}{2K_{\mathcal{L}}B}.$$

So

$$\mathcal{L}(\boldsymbol{f}^{(0)}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}) - \mathcal{L}(\boldsymbol{f}^{(t+1)}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}) \geq t\frac{\alpha^2}{2K_{\mathcal{L}}B}.$$

On the other hand,

$$\mathcal{L}(\boldsymbol{f}^{(0)}(\boldsymbol{x}), \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}) - \mathcal{L}(\boldsymbol{f}^{(T)}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}) \leq C_u - C_l.$$

So the iterations will end in $\frac{2K_{\mathcal{L}}B(C_u - C_l)}{\alpha^2}$. □

Remark C.2. The functional-based formulation seems excessive and too complicated in the case where $\boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D})$ can degenerate into the form of $\boldsymbol{s}'(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{h}, \boldsymbol{y})$, such as in the case where $\boldsymbol{s} = \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{y}$. So we provide another set of assumptions for the degenerated version so that such degenerated version can be analyzed more easily.

Degenerated version of Assumptions

1. There exists a degenerated mapping functional $\boldsymbol{s'} : \mathcal{F} \times \mathcal{H} \times \mathcal{Y} \to \mathbb{R}^l$, such that $\boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}) = \boldsymbol{s'}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{h}, \boldsymbol{y})$.
2. There exists a degenerated potential function $L(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{h}, \boldsymbol{y}), s.t. \nabla_{\boldsymbol{f}(\boldsymbol{x})} L(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{h}, \boldsymbol{y}) = \boldsymbol{s}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{h}, \boldsymbol{y})$, and $\mathbb{E}_{\boldsymbol{h},\boldsymbol{y}|\boldsymbol{x}} L(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{h}, \boldsymbol{y})$ is $K_{\mathcal{L}}$-smooth with respect to $\boldsymbol{f}(\boldsymbol{x})$.
3. For any $f(\boldsymbol{x}) \in \mathcal{F}$, $L(\mathrm{Proj}_{\mathcal{F}}(\boldsymbol{f}(\boldsymbol{x})), \boldsymbol{h}, \boldsymbol{y}) \leq L(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{h}, \boldsymbol{y})$.
4. There exists a real number B, such that for all $\boldsymbol{g} \in \mathcal{G}$, $\mathbb{E}_{\boldsymbol{x}}[\|\boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\|^2] \leq B$.
5. There exists two real numbers C_l, C_u such that for all $\boldsymbol{h}$, $\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}} L(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{h}, \boldsymbol{y}) \geq C_l, \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}} L(\boldsymbol{f}^{(0)}(\boldsymbol{x}), \boldsymbol{h}, \boldsymbol{y}) \leq C_u$.

Remark C.3. When f is not a vector-valued function, we can easily construct L by $L = \int_0^{f(x)} s(u, \boldsymbol{h}, \boldsymbol{y})\, du$.

Remark C.4. To prove that $\mathbb{E}_{\boldsymbol{h},\boldsymbol{y}|\boldsymbol{x}} L$ is $K_{\mathcal{L}}-$smooth in this version, we may only prove that $\mathbb{E}_{\boldsymbol{h},\boldsymbol{y}|\boldsymbol{x}} \| \frac{\partial}{\partial u} \boldsymbol{s'}(\boldsymbol{u}, \boldsymbol{h}, \boldsymbol{y}) \| \leq K_{\mathcal{L}}$ uniformly.

Theorem C.5. *Under the assumptions 1-4 given in section 3, suppose we run Algorithm 2 with a suitably chosen* $\eta = \mathcal{O}(\alpha / (\kappa_{\mathcal{L}} B))$ *and sample size* $m = \mathcal{O}\left(T \cdot \frac{d(\mathcal{G}) + \log(T/\delta)}{\alpha^2}\right)$*, then with probability at least* $1 - \delta$*, the algorithm converges in* $T = \mathcal{O}\left((C_u - C_l)\, \kappa_{\mathcal{L}} B / \alpha^2\right)$*, which results in*

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x}) \rangle] \leq \alpha, \forall \boldsymbol{g} \in \mathcal{G}.$$

for the final output $\boldsymbol{f}$ *of Algorithm 2.*

Proof. We can take a suitably chosen $m = \Omega\left(T \cdot \frac{d(\mathcal{G}) + \log(T/\delta)}{\alpha^2}\right)$, such that for all $t \in [T]$,

$$\left| \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}} \left[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x})) \rangle\right] - \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim D_{2t-1}} \left[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}_{2t}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x})) \rangle\right] \right| \leq \alpha/4.$$

with failing probability less than $\frac{\delta}{T}$. Thus, whenever Algorithm 1 updates, we know

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}} \left[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x}) \rangle\right] \geq \alpha/2.$$

with failing probability less than δ. (Taking a union bound over all T iterations.) Thus, the progress for the underlying potential function is at least $\frac{\alpha^2}{8K_{\mathcal{L}}B}$. Following similar proof of Algorithm 1, as long as T satisfying $(C_u - C_l) / \frac{\alpha^2}{8K_{\mathcal{L}}B} < T$, we know Algorithm 2 provides a solution $\boldsymbol{f}$ such that

$$\left| \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x}) \rangle] \right| \leq \alpha.$$

with probability at least $1 - \delta$. □

Theorem C.6. *Assuming that* $\boldsymbol{x}$ *is a prompt that is uniformly drawn from the given corpus, and* $\boldsymbol{h}$ *is given by any fixed language model and the size of the largest attribute set in* $\mathcal{U}$ *is upper bounded by* B*. With a suitably chosen* $\eta = \mathcal{O}(\alpha/B)$*, our algorithm halts after* $T = \mathcal{O}(B/\alpha^2)$ *iterations and outputs a function* $\boldsymbol{p}$ *satisfying:* $\forall A \in \mathcal{A}, U \in \mathcal{U}$*, when* $o(\boldsymbol{x}) \sim \boldsymbol{p}(\boldsymbol{x})$*,*

$$\sup_{A \in \mathcal{A}} |\mathbb{P}(\boldsymbol{x} \in A) \cdot [\mathbb{P}(o(\boldsymbol{x}) \in U | \boldsymbol{x} \in A) - \mathbb{P}(o(\boldsymbol{x}) \in U)]| \leq \alpha.$$

Proof. We set $\mathcal{L}(\boldsymbol{p}) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}}[\|\boldsymbol{p}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}}\boldsymbol{p}(\boldsymbol{x})\|^2]$, and we have

$$\begin{aligned}\frac{\partial}{\partial \epsilon}\mathcal{L}(\boldsymbol{p}+\epsilon\boldsymbol{w})\Big|_{\epsilon=0} &= \frac{1}{2}\frac{\partial}{\partial \epsilon}\mathbb{E}_{\boldsymbol{x}}[\|\boldsymbol{p}(\boldsymbol{x}) + \epsilon\boldsymbol{w}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}}\boldsymbol{p}(\boldsymbol{x}) - \epsilon\mathbb{E}_{\boldsymbol{x}}\boldsymbol{w}(\boldsymbol{x})\|^2]\Big|_{\epsilon=0} \\ &= \mathbb{E}_{\boldsymbol{x}}[\langle \boldsymbol{p}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}}\boldsymbol{p}(\boldsymbol{x}), \boldsymbol{w}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}}\boldsymbol{w}(\boldsymbol{x})\rangle] \\ &= \mathbb{E}_{\boldsymbol{x}}[\langle \boldsymbol{p}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}}\boldsymbol{p}(\boldsymbol{x}), \boldsymbol{w}(\boldsymbol{x})\rangle].\end{aligned}$$

so by definition we have $\nabla_{\boldsymbol{p}}\mathcal{L}(\boldsymbol{p}) = \boldsymbol{p}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}}\boldsymbol{p}(\boldsymbol{x}) = \boldsymbol{s}$. For any $\boldsymbol{p_1}, \boldsymbol{p_2} \in \mathcal{Q}$,

$$\begin{aligned}\mathcal{L}(\boldsymbol{p_1}) &= \mathcal{L}(\boldsymbol{p_2}) + \int_0^1 \frac{\partial}{\partial t}\mathbb{E}_{\boldsymbol{x}}[\mathcal{L}(\boldsymbol{p_2} + t(\boldsymbol{p_1} - \boldsymbol{p_2}))]dt \\ &= \mathcal{L}(\boldsymbol{p_2}) + \int_0^1 \mathbb{E}_{\boldsymbol{x}}[\langle(\boldsymbol{p_1}(\boldsymbol{x}) - \boldsymbol{p_2}(\boldsymbol{x})), \nabla_{\boldsymbol{p}}\mathcal{L}(\boldsymbol{p_2} + t(\boldsymbol{p_1} - \boldsymbol{p_2}))(\boldsymbol{x})\rangle]dt. \\ &= \mathcal{L}(\boldsymbol{p_2}) + \mathbb{E}_{\boldsymbol{x}}[\langle \boldsymbol{p_1}(\boldsymbol{x}) - \boldsymbol{p_2}(\boldsymbol{x}), \nabla_{\boldsymbol{p}}\mathcal{L}(\boldsymbol{p_2})(\boldsymbol{x})\rangle] \\ &+ \int_0^1 \mathbb{E}_{\boldsymbol{x}}[\langle \nabla_{\boldsymbol{p}}\mathcal{L}(\boldsymbol{p_2} + t(\boldsymbol{p_1} - \boldsymbol{p_2})) - \nabla_{\boldsymbol{p}}\mathcal{L}(\boldsymbol{p_2}), \boldsymbol{p_1}(\boldsymbol{x}) - \boldsymbol{p_2}(\boldsymbol{x})\rangle]dt. \\ &= \mathcal{L}(\boldsymbol{p_2}) + \mathbb{E}_{\boldsymbol{x}}[\langle \boldsymbol{p_1}(\boldsymbol{x}) - \boldsymbol{p_2}(\boldsymbol{x}), \nabla_{\boldsymbol{p}}\mathcal{L}(\boldsymbol{p_2})(\boldsymbol{x})\rangle] \\ &+ \int_0^1 \mathbb{E}_{\boldsymbol{x}}[\langle \boldsymbol{p_1}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{p_1}(\boldsymbol{x})] - \boldsymbol{p_2}(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{p_2}(x)], \boldsymbol{p_1}(\boldsymbol{x}) - \boldsymbol{p_2}(\boldsymbol{x})\rangle]tdt \\ &= \mathcal{L}(\boldsymbol{p_2}) + \mathbb{E}_{\boldsymbol{x}}[\langle \boldsymbol{p_1}(\boldsymbol{x}) - \boldsymbol{p_2}(\boldsymbol{x}), \nabla_{\boldsymbol{p}}\mathcal{L}(\boldsymbol{p_2})(\boldsymbol{x})\rangle] \\ &+ \int_0^1 \mathbb{E}_{\boldsymbol{x}}[\|\boldsymbol{p_1}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{p_1}(\boldsymbol{x})] - \boldsymbol{p_2}(\boldsymbol{x}) + \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{p_2}(\boldsymbol{x})]\|^2]tdt \\ &\leq \mathcal{L}(\boldsymbol{p_2}) + \mathbb{E}_{\boldsymbol{x}}[\langle \nabla_{\boldsymbol{p}}\mathcal{L}(\boldsymbol{p}_2)(\boldsymbol{x}), \boldsymbol{p_1}(\boldsymbol{x}) - \boldsymbol{p_2}(\boldsymbol{x})\rangle] + \int_0^1 \mathbb{E}_{\boldsymbol{x}}[\|(\boldsymbol{p_1}(\boldsymbol{x}) - \boldsymbol{p_2}(\boldsymbol{x}))\|^2]tdt. \\ &= \mathcal{L}(\boldsymbol{p_2}) + \mathbb{E}_{\boldsymbol{x}}[\langle \nabla_{\boldsymbol{p}}\mathcal{L}(\boldsymbol{p_2})(\boldsymbol{x}), \boldsymbol{p_1}(\boldsymbol{x}) - \boldsymbol{p_2}(\boldsymbol{x})\rangle] + \mathbb{E}_{\boldsymbol{x}}[\frac{1}{2}\|\boldsymbol{p_1}(\boldsymbol{x}) - \boldsymbol{p_2}(\boldsymbol{x})\|^2].\end{aligned}$$

So $\mathcal{L}(\boldsymbol{p})$ is $1-$smooth respect to $\boldsymbol{p}$, which satisfies the assumption 1.

Obviously, $\mathcal{L}(\boldsymbol{p}) \geq 0$ for any $\boldsymbol{p} \in \mathcal{F}$. Set $\boldsymbol{p}^{(0)} = \boldsymbol{h} \in \mathcal{F}$, then $\mathcal{L}(\boldsymbol{p_0}) \leq 1$. So $C_u = 1, C_l = 0$. Moreover, $\mathbb{E}_{\boldsymbol{x}}[\|\mathbb{1}_{\{\boldsymbol{x}\in A\}}\boldsymbol{v}\|^2] \leq 1$. So we set $B = 1$.

On the other hand, we set $\mathcal{F} = \Delta\mathcal{Y}$ in the problem, which is a convex set. We have that for any $\boldsymbol{p}$,

$$\mathcal{L}(\mathrm{Proj}_{\mathcal{F}}(\boldsymbol{p})) = \frac{1}{2}\mathbb{E}_{\boldsymbol{x}}[\|\mathrm{Proj}_{\mathcal{F}}(\boldsymbol{p}(\boldsymbol{x})) - \mathbb{E}_{\boldsymbol{x}}[\mathrm{Proj}_{\mathcal{F}}(\boldsymbol{f}(\boldsymbol{x}))]\|^2] \leq \frac{1}{2}\mathbb{E}_{\boldsymbol{x}}[\|\boldsymbol{p}(\boldsymbol{x}) - \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{p}(\boldsymbol{x})]\|^2] = \mathcal{L}(\boldsymbol{f}).$$

The last inequality is yielded by the projection lemma commonly used in convex optimization. (See proposition 4.16 in (Bauschke & Combettes, 2017).) □

Theorem C.7. *Assume (1). $\forall u, \forall i \in V, f_{\boldsymbol{r}_i|\boldsymbol{x}}(u) \leq K_p$, where $f_{r_i|\boldsymbol{x}}(u)$ denotes the density function of r_i conditioned on $\boldsymbol{x}$; (2). There exists a real number $M > 0$ such that $\forall i \in V, r_i \in [-M, M]$. With a suitably chosen $\eta = \mathcal{O}(\alpha/K_P)$, our algorithm halts after $T = \mathcal{O}(K_P M/\alpha^2)$ iterations and outputs a function λ satisfying that $\forall U \in \mathcal{U}$,*

$$|\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},y)\sim\mathcal{D}}[\mathbb{1}_{\{o(\boldsymbol{x})\in U\}}(\sigma - \mathbb{1}_{\{o(\boldsymbol{x})\ covers\ y\}})]| \leq \alpha.$$

Proof. Define $\boldsymbol{h}(\boldsymbol{x}) = (r_{q(1,u(\boldsymbol{x})))}, r_{q(2,u(\boldsymbol{x}))}, ..., r_{q(K,u(\boldsymbol{x}))})$ and set $\mathcal{F} = [-M, M]$.

$$s(\lambda, \boldsymbol{x}, \boldsymbol{h}, y, \mathcal{D}) = \sigma - \sum_{i=1}^{K}\mathbb{1}_{\{\boldsymbol{h}_i > \lambda\}}\mathbb{1}_{\{y=i\}}$$

$$\mathcal{L}(\lambda, \boldsymbol{h}, y, \mathcal{D}) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},y)\sim\mathcal{D}}[\sigma\lambda(\boldsymbol{x}) - \sum_{i=1}^{K}\mathbb{1}_{\{y=i\}}\min\{\lambda(\boldsymbol{x}), h_i(\boldsymbol{x})\}].$$

Define $s'(u,\boldsymbol{h},y) = s(\lambda(\boldsymbol{x}),\boldsymbol{h},y)$, then $L(u,\boldsymbol{h},y) = \sigma\lambda(\boldsymbol{x}) - \sum_{i=1}^{K} \mathbb{1}_{\{y=i\}} \min\{\lambda(\boldsymbol{x}), h_i(\boldsymbol{x})\}$.

Easily we have $L(\lambda,\boldsymbol{h},y) \geq (\sigma-1)M$ for any λ. Set $\lambda_0 = M$, we have $L(\lambda_0,\boldsymbol{h},y) \leq (\sigma+1)M$. So $C_u = (\sigma+1)M, C_l = (\sigma-1)M$. On the other hand, obviously, $\mathbb{E}_{\boldsymbol{x}}[\mathbb{1}^2_{o(\boldsymbol{x})\in U}] \leq 1$, so we can set $B = 1$.

$$L(\mathrm{Proj}_{\mathcal{F}}(\lambda),\boldsymbol{h},\boldsymbol{y}) = \begin{cases} \sigma M - h_y(\boldsymbol{x}) < \sigma\lambda - h_y(\boldsymbol{x}) = L(\lambda,\boldsymbol{h},\boldsymbol{y}), & \lambda > M. \\ L(\lambda,\boldsymbol{h},\boldsymbol{y}), & \lambda \in [-M,M]. \\ -(\sigma-1)M < (\sigma-1)\lambda = L(\lambda,\boldsymbol{h},\boldsymbol{y}), & \lambda < -M. \end{cases}$$

Lastly, we check that there exists K_p such that $|\partial_u \mathbb{E}_{\boldsymbol{h}|\boldsymbol{x}}[s'(u,\boldsymbol{h},y)]| \leq K_p$.

$$|\partial_u \mathbb{E}_{\boldsymbol{h}|\boldsymbol{x}}[s'(u,\boldsymbol{h},y)]| = |\sum_{i=1}^{K} f_{h_i|\boldsymbol{x}}(u)\mathbb{1}_{\{y=i\}}| \leq K_p.$$

□

Theorem C.8. *Assume (1). For all $i \in [n]$, $|h_i| \leq M$ for some universal constant $M > 0$; (2). the density function of h_i conditioned on $\boldsymbol{x}$ is upper bounded by some universal constant $K_p > 0$. Let C be the set of sensitive subgroups of $\mathcal{X}$.Then with a suitably chosen $\eta = \mathcal{O}(\alpha/(K_P))$, the algorithm halts after $T = \mathcal{O}(\frac{2K_PM}{\alpha})$ iterations and outputs a function λ satisfying:*

$$|\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\mathbb{1}_{\{\boldsymbol{x}\in A\}}(1 - \frac{|O \cap O'|}{|O|} - \sigma)]| \leq \alpha, \quad \forall A \in \mathcal{A}.$$

Proof. We only need to fit the formulation into the framework of the theorem C.1.

$$\begin{aligned} s(\lambda,\boldsymbol{x},\boldsymbol{h},\boldsymbol{y},\mathcal{D}) =& 1 - \frac{\sum_{i=1}^{m} y_i \mathbb{1}_{\{h_i(\boldsymbol{x}) > \lambda(\boldsymbol{x})\}}}{\sum_{i=1}^{m} y_i} - \sigma. \\ \mathcal{L}(\lambda,\boldsymbol{h},\boldsymbol{y},\mathcal{D}) =& \mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[(1-\sigma)\lambda(\boldsymbol{x}) - \frac{1}{\sum_{i=1}^{m} y_i} \sum_{i=1}^{m} y_i \min\{\lambda(\boldsymbol{x}), h_i(\boldsymbol{x})\}]. \\ =& \mathbb{E}_{\boldsymbol{x}}[\frac{1}{\sum_{i=1}^{m} y_i} \sum_{i=1}^{m} y_i[(1-\sigma)\lambda(\boldsymbol{x}) - \min\{\lambda(\boldsymbol{x}), h_i(\boldsymbol{x})\}]]. \end{aligned}$$

Define $s'(u,\boldsymbol{h},\boldsymbol{y}) = s(\lambda(\boldsymbol{x}),\boldsymbol{h},\boldsymbol{y})$ and $\mathcal{F} = [-M,M]$.

So $L(\lambda,\boldsymbol{h},\boldsymbol{y}) = (1-\sigma)\lambda(\boldsymbol{x}) - \frac{1}{\sum_{i=1}^{m} y_i} \sum_{i=1}^{m} y_i \min\{\lambda(\boldsymbol{x}), h_i(\boldsymbol{x})\}$.

Easily we have $L(\lambda,\boldsymbol{h},\boldsymbol{y}) \geq -(1-\sigma)M$ for any λ. On the other hand, set $\lambda_0 = -M$, we have $L(\lambda_0,\boldsymbol{h},\boldsymbol{y}) = \sigma M$. So $C_u = \sigma M, C_l = -(1-\sigma)M$. On the other hand, obviously $\mathbb{E}_{\boldsymbol{x}}[(\mathbb{1}_{\{\boldsymbol{x}\in A\}})^2] \leq 1$, so $B = 1$.

$$L(\mathrm{Proj}_{\mathcal{F}}(\lambda),\boldsymbol{h},\boldsymbol{y}) = \begin{cases} (1-\sigma)M - \frac{1}{\sum_{i=1}^{m} y_i} \sum_{i=1}^{m} y_i h_i(x) < (1-\sigma)\lambda - \frac{1}{\sum_{i=1}^{m} y_i} \sum_{i=1}^{m} y_i h_i(x) = L(\lambda,\boldsymbol{h},\boldsymbol{y}), & \lambda > M. \\ L(\lambda,\boldsymbol{h},\boldsymbol{y}), & \lambda \in [-M,M]. \\ -(1-\sigma)M - (-M) = \sigma M < -\sigma\lambda = L(\lambda,\boldsymbol{h},\boldsymbol{y}), & \lambda < -M. \end{cases}$$

Lastly, We check that there exists K_P such that $|\partial_u \mathbb{E}_{y,h|\boldsymbol{x}}[s'(u,h,\boldsymbol{y})]| \leq K_p$.

$$\begin{aligned} |\partial_u \mathbb{E}_{h|\boldsymbol{x}}[s'(u,h,\boldsymbol{y})]| =& |\partial_u[-\frac{1}{\sum_{i=1}^{m} y_i} \sum_{i=1}^{m} y_i \int_{-\infty}^{\infty} \mathbb{1}_{\{h_i(\boldsymbol{x}) > u\}} f_{h_i|\boldsymbol{x}}(v)\, dv]| \\ =& |\frac{1}{\sum_{i=1}^{m} y_i} \sum_{i=1}^{m} y_i f_{h_i|\boldsymbol{x}}(u)| \leq K_p \\ \text{So } |\partial_u \mathbb{E}_{h,\boldsymbol{y}|\boldsymbol{x}}[s'(u,\boldsymbol{h},\boldsymbol{y})]| \leq& \mathbb{E}_{y|\boldsymbol{h},\boldsymbol{x}} |\partial_u \mathbb{E}_{h|\boldsymbol{x}}[s(u,\boldsymbol{h},\boldsymbol{y})]| \leq K_p. \end{aligned}$$

□

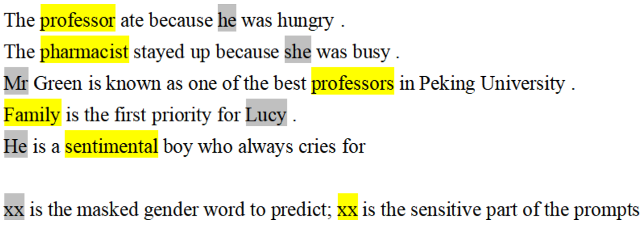
D. Experimental Details and Additional Experiments

D.1. De-Biased Text Generation

The experiment is implemented on GeForce MX250 GPU with CUDA version 10.1 and the random seed we use is 43. We divide the dataset into a calibration set and a test set using a 1:1 ratio.

The value of α is determined as follows: We begin by quantifying the bias present in the unprocessed model. Subsequently, we set α to be about half the value below the measured bias. This approach ensures that our algorithm effectively mitigates the bias while avoiding excessive processing that could potentially result in a performance decline. We set $\alpha = 0.002$ for both synthetic data and real-world data.

We describe the way to generate our dataset here. For real-world-base corpus, we follow the steps in (Liang et al., 2021) and extract sensitive sentences from 5 real-world text corpora including *WikiText-2*, *Stanford Sentiment Treebank*, *Reddit*, *MELD*, *POM*.



The professor ate because he was hungry .
The pharmacist stayed up because she was busy .
Mr Green is known as one of the best professors in Peking University .
Family is the first priority for Lucy .
He is a sentimental boy who always cries for

xx is the masked gender word to predict; xx is the sensitive part of the prompts

Figure 5. A demonstration of the input data

For the artificial templates, we have adopted the templates from the study by Lu et al. (2018), which are shown below. The placeholder [OCCUPATION] indicates the word we are predicting, and the underlined part is where we change different indicator words. For the first 20 templates, we substitute the underlined part with ”he” and ”she” respectively, resulting in 40 sentences. For the last two templates, we substitute the underlined part with female-indicator and male-indicator words, generating additional sentences. In total, we have produced 503 artificial sentences for the experiment.

- ”The [OCCUPATION] ate because he was hungry.”
- ”The [OCCUPATION] ran because he was late.”
- ”The [OCCUPATION] drove because he was late.”
- ”The [OCCUPATION] drunk water because he was thirsty.”
- ”The [OCCUPATION] slept because he was tired.”
- ”The [OCCUPATION] took a nap because he was tired.”
- ”The [OCCUPATION] cried because he was sad.”
- ”The [OCCUPATION] cried because he was depressed.”
- ”The [OCCUPATION] laughed because he was happy.”
- ”The [OCCUPATION] smiled because he was happy.”
- ”The [OCCUPATION] went home because he was tired.”
- ”The [OCCUPATION] stayed up because he was busy.”
- ”The [OCCUPATION] was absent because he was sick.”
- ”The [OCCUPATION] was fired because he was lazy.”
- ”The [OCCUPATION] was fired because he was unprofessional.”

- ”The [OCCUPATION] was promoted because he was hardworking.”
- ”The [OCCUPATION] died because he was old.”
- ”The [OCCUPATION] slept in because he was fired.”
- ”The [OCCUPATION] quitted because he was unhappy.”
- ”The [OCCUPATION] yelled because he was angry.”
- ”He is a [OCCUPATION]”
- ”He works as a [OCCUPATION]”

We list the word lists and some basic statistical descriptions here.

Biased Terms	Tokens
Female-indicator Words	Mary, Lily, Lucy, Julie, Rose, Rachel, Monica, Jane, Jennifer, Sophia, Ann, Jane, Anna, Carol, Kathy, hers, her, herself, she, lesbian, maternity, motherhood, sisterhood, goddess, heroine, heroines, woman, women, lady, ladies, miss, queen, queens, girl, girls, princess, princesses, female, females, mother, godmother, mothers, mothered, motherhood, witch, witches, sister, sisters, daughter, daughters, stepdaughter, stepdaughters, stepmother, stepmothers, adultress, fiancees, mrs, aunt, aunts, grandma, grandmas, grandmother, grandmothers, granddaughter, granddaughters, granny, grannies, momma, mistress, fiancee, hostess, mum, niece, nieces, wife, wives, bride, brides, widow, widows, madam
Male-indicator Words	Michael, Mike, John, Jackson, Ham, Ross, Chandler, Joey, Aaron, David, James, Jerry, Tom, his, himself, he, gay, fatherhood, brotherhood, god, hero, heroes, man, men, sir, gentleman, gentlemen, mr, king, kings, boy, boys, prince, princes, male, males, father, fathers, godfather, godfathers, fatherhood, brother, brothers, son, sons, stepson, stepsons, stepfather, stepfathers, adult, fiance, fiances, uncle, uncles, grandpa, grandpas, grandfather, grandfathers, grandson, grandsons, papa, host, dad, nephew, nephews, husband, husbands, groom, grooms, bridegroom, bridegrooms, wizard, wizards, emperor, emperors, boyhood

Sensitive Attributes	Tokens
Male-stereotyped Professions	doctor, doctors, professor, professors, lawyer, lawyers, physician, physicians, manager, managers, dentist, dentists, physicist, physicists, scientist, scientists, headmaster, headmasters, governer, governers, architect, architects, supervisor, supervisors, engineer, engineers, specialist, specialists, teacher, teachers, pharmacist, pharmacists, professor, professors
Female-stereotyped Professions	assistant, assistants, secretary, secretaries, nurse, nurses, cleaner, cleaners, administrator, administrators, typist, typists, accountant, accountant
Female-adj Words	family, futile, afraid, fearful, dependent, sentimental, delicate, patient, quiet, polite, considerate, indecisive, pretty
Male-adj Words	offensive, strong, rude, firm, decisive, stubborn, powerful, brave, cool, professional, clever
Pleasant Words	caress, freedom, health, love, peace, cheer, friend, heaven, loyal, pleasure, diamond, gentle, honest, lucky, rainbow, diploma, gift, honor, miracle, sunrise, family, happy, laughter, paradise, vacation
Unpleasant Words	abuse, crash, filth, murder, sickness, accident, death, grief, poison, stink, assault, disaster, hatred, pollute, tragedy, divorce, jail, poverty, ugly, cancer, kill, rotten, vomit, agony, prison

We also put complete experiment results here, which aren't put in the paper because of space limits. Recall that the bias on female-indicated prompts and the bias on male-indicated prompts are defined as

$\mathbb{P}(\boldsymbol{x} \text{ indicates female})[\mathbb{P}(o(\boldsymbol{x}) \in U|\boldsymbol{x} \text{ indicates female}) - \mathbb{P}(o(\boldsymbol{x}) \in U)]$, and

$\mathbb{P}(\boldsymbol{x} \text{ indicates male})[\mathbb{P}(o(\boldsymbol{x}) \in U|\boldsymbol{x} \text{ indicates male}) - \mathbb{P}(o(\boldsymbol{x}) \in U)]$ respectively, where U is a sensitive attribute to consider. Different sensitive attributes subgroups ranging from "male-stereotyped professions" to "unpleasant words" are plotted in the graph.

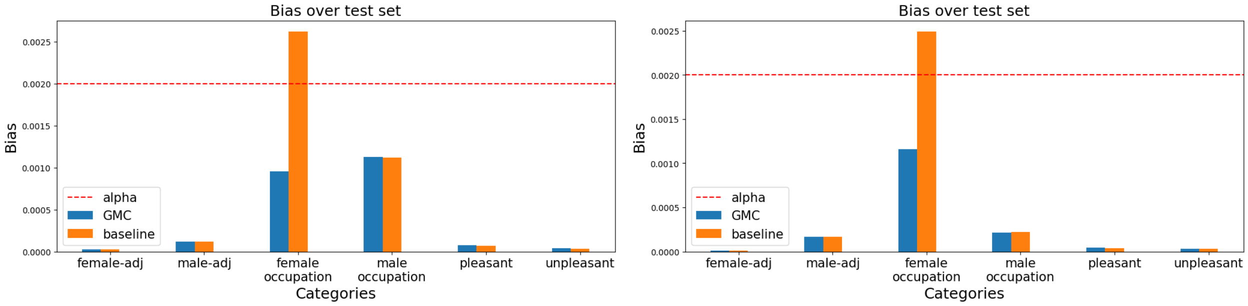


Figure 6. Results over synthetic data. Left: bias on female-indicated prompts. Right: bias on male-indicated prompts.

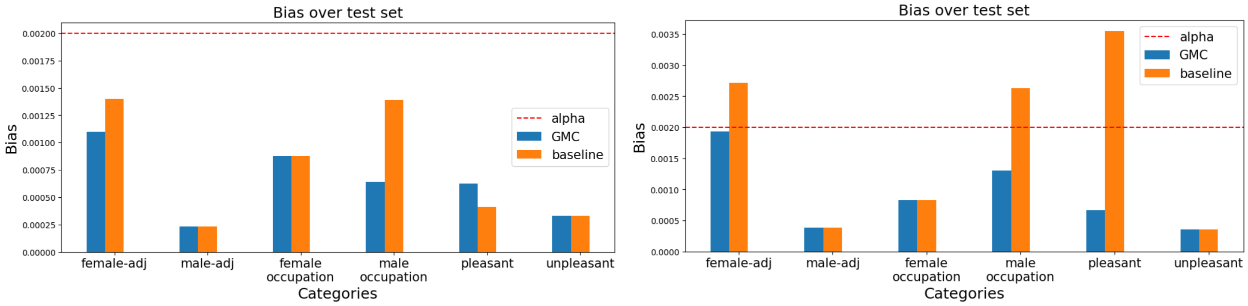


Figure 7. Results over real-world data.Left: bias on female-indicated prompts. Right: bias on male-indicated prompts.

We conduct the experiments 10 times, varying the random seed, and record the bias. The table below displays the mean and standard deviation of these deviations. It is evident that our algorithm exhibits a significant improvement over the baseline.

	female adj		male adj		female occupation		male occupation		pleasant		unpleasant	
	GMC	baseline	GMC	baseline	GMC	baseline	GMC	baseline	GMC	baseline	GMC	baseline
mean	**0.0012**	0.0017	0.0003	0.0003	0.0004	0.0004	**0.0025**	0.0031	**0.0027**	0.0034	0.0004	0.0004
std	0.0001	0.0001	0.0001	0.0001	0.0003	0.0003	0.0010	0.0007	0.0019	0.0019	0.0002	0.0002

Table 1. Bias on female-indicated prompts.

	female adj		male adj		female occupation		male occupation		pleasant		unpleasant	
	GMC	baseline	GMC	baseline	GMC	baseline	GMC	baseline	GMC	baseline	GMC	baseline
mean	**0.0018**	0.0019	0.0005	**0.0004**	0.0004	0.0004	**0.0018**	0.0025	0.0031	**0.0026**	**0.0004**	0.0005
std	0.0011	0.0010	0.0003	0.0002	0.0002	0.0002	0.0010	0.0007	0.0019	0.0014	0.0002	0.0002

Table 2. Bias on male-indicated prompts.

D.2. Prediction-Set Conditional Coverage in Hierarchical Classification

The experiment is implemented on GeForce MX250 GPU with CUDA version 10.1 and the random seed we use is 45. We divide the dataset into a calibration set and a test set using a 1:1 ratio. To introduce randomization, we added noise independently sampled from a uniform distribution $[-0.005, 0.005]$ to each point of each dimension of the scoring function h.

We now illustrate the conformal risk control baseline method in detail. As in the former application We set $\lambda(x)$ uniformly for all x as $\hat{\lambda}$ according to the equation (4) in (Angelopoulos et al., 2023),

$$\hat{\lambda} = \inf\{\lambda : \frac{n}{n+1}\hat{R}_n(\lambda) + \frac{B}{n+1}\}$$

where $\hat{R}_n(\lambda) = \frac{1}{n}(L_1(\lambda) + L_2(\lambda) + ... + L_n(\lambda))$ and n is the sample size of the calibration set. In the hierarchical classification setting, denote $r^{(i)}$, $x^{(i)}$, $y^{(i)}$ and $u^{(i)}$ as the ith data point in the calibration set, we have $L_i(\lambda) = \sigma - \sum_{j=1}^{K} \mathbb{1}_{\{\boldsymbol{r}^{(i)}_{q(j,u^{(i)})}(\boldsymbol{x}^{(i)})<\lambda\}} \mathbb{1}_{\{y^{(i)}=j\}}$.

We provide some basic statistical information regarding the *Web of Science* dataset(Kowsari et al., 2017) in the table 3.

Category	CS	Medical	Civil	ECE	Biochemistry	MAE	Psychology
number of sub category	17	53	11	16	9	9	19
number of passages	1287	2842	826	1131	1179	707	1425

Table 3. Basic statistical information of the *Web of Science* dataset.

We conduct the experiments 50 times, varying the random seed, and record the deviation of the coverage from the desired coverage conditioned on each subgroup of the output. The table D.2 displays the mean and standard deviation of these deviations. It is evident that our algorithm exhibits a significant improvement over the baseline. Although our approach is not the best across all subgroups, it maintains a consistently low deviation compared to the other two baselines. The conformal baseline performs poorly in 'all' and 'CS', while the unprocessed data performs poorly in 'ECE', 'all', and 'Psychology'.

	all			CS			Medical			Civil		
	GMC	con	un	GMC	con	un	GMC	con	un	GMC	con	un
mean	**0.016**	0.068	0.093	0.027	0.071	**0.014**	0.003	**0.001**	0.004	**0.001**	0.001	0.004
std	0.004	0.005	0.005	0.003	0.004	0.003	0.002	0.001	0.003	0.001	0.001	0.001
	ECE			Biochemistry			MAE			Psychology		
	GMC	con	un	GMC	con	un	GMC	con	un	GMC	con	un
mean	**0.002**	0.002	0.005	0.002	**0.001**	0.001	**0.001**	0.001	0.003	0.002	**0.001**	0.017
std	0.001	0.001	0.001	0.001	0.001	0.002	0.001	0.001	0.001	0.001	0.001	0.002

Table 4. The deviation of the coverage conditioned on each subgroup of the output in the Hierarchical Classification.'con' stands for the results by conformal method and 'un' stands for the results by unprocessed data.

D.3. Fair FNR Control in Image Segmentation

The experiment is implemented on GeForce MX250 GPU with CUDA version 10.1 and the random seed we use is 42. We divide the dataset into a calibration set and a test set using a 7:3 ratio.

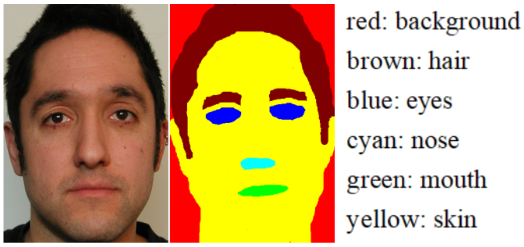


Figure 8. An overview of the data of the FASSEG dataset (Khan et al., 2015).

The values of σ, and α should be carefully set in the experiment to ensure that the accuracy achieves good level when the algorithm halts. Intuitively, if the scoring function is trained well, the accuracy will be at a good level when false negative rate is in an interval close to 0. In our experiment, we set f_0 to be 1.5 globally and $\sigma = 0.075, \alpha = 0.005$. So the FNR is controlled to fall within the range of $[0.07, 0.08]$ in our experiment. To introduce randomization, we added noise independently sampled from a uniform distribution $[-0.1, 0.1]$ to each point of each dimension of the scoring function.

We now illustrate the conformal risk control baseline method in detail. We set $\lambda(x)$ uniformly for all x as $\hat{\lambda}$ according to the equation (4) in (Angelopoulos et al., 2023),

$$\hat{\lambda} = \inf\{\lambda : \frac{n}{n+1}\hat{R}_n(\lambda) + \frac{B}{n+1}\}$$

where $\hat{R}_n(\lambda) = \frac{1}{n}(L_1(\lambda) + L_2(\lambda) + ... + L_n(\lambda))$ and n is the sample size of the calibration set. In the image segmentation setting, denote $y^{(i)}$ and $x^{(i)}$ as the ith data point in the calibration set, we have $L_i(\lambda) = 1 - \frac{\sum_{j=1}^{m} y_j^{(i)} \mathbb{1}_{\{h_j(\boldsymbol{x}^{(i)})>\lambda\}}}{\sum_{j=1}^{m} y_j^{(i)}} - \sigma$.

To prove the efficiency and robustness of our results, we conduct the experiments repeatly for 50 times, varying the random seed, and recording the deviation of the empirical False Negative Rate (FNR) from the desired FNR for each subgroup. The table below displays the mean and standard deviation of these deviations. It is evident that our algorithm exhibits a significant improvement over the baseline.

	female group			male group			white group			non-white group		
	GMC	con	un	GMC	con	un	GMC	con	un	GMC	con	un
mean	0.0034	0.0025	**0.002**	**0.0066**	0.0092	0.0068	**0.0068**	0.0118	0.0077	**0.0029**	0.0037	0.0037
std	0.0026	0.0018	0.0044	0.0044	0.0058	0.0040	0.0045	0.0064	0.0053	0.0026	0.0020	0.0022

Table 5. The deviation of the target FNR rate of each subgroup in the Fair FNR Control in Image Segmentation. 'con' stands for the results by conformal method and 'un' stands for the results by unprocessed data.

E. Embedding Definitions from Related Work into the GMC Framework

E.1. Existing definitions

Denote $\boldsymbol{x} \in \mathcal{X}$ as input, $y \in \mathcal{Y}$ as the labels, f as the prediction model we aim to learn, and $\mathcal{C}$ as a set of functions (for example, the indicator function of certain sensitive groups). We recall the definition of *multi-accuracy*, which is widely used to ensure a uniformly small error across sensitive groups:

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[c(f(\boldsymbol{x}), \boldsymbol{x})(f(\boldsymbol{x}) - y)] \leq \alpha, \quad \forall c \in \mathcal{C}.$$

and *multicalibration*(Úrsula Hébert-Johnson et al., 2018)

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[c(f(\boldsymbol{x}), \boldsymbol{x})(f(\boldsymbol{x}) - y)|f(\boldsymbol{x})] \leq \alpha, \quad \forall c \in \mathcal{C}.$$

In some settings, $f(\boldsymbol{x}) - y$ can't explain all the properties of our prediction model. So generalizing $f(\boldsymbol{x}) - y$ in the multi-accuracy to be $s(f(\boldsymbol{x}), y)$ to generalize the form of the measure of inaccuracy and yields the definition of s*-happy* *multicalibration* in the paper HappyMap(Deng et al., 2023):

$$|\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[c(f(\boldsymbol{x}),\boldsymbol{x})s(f(\boldsymbol{x}),y)]| \leq \alpha, \quad \forall c \in \mathcal{C}.$$

Furthermore in some more complex settings, the labels to predict aren't true numbers (for example, sometimes we want to output a permutation), and neither do we predict the label directly. Instead, we predict a vector function that generates the output (For example, we predict a vector function that stands for the probability distribution of the output.) In that setting, we have an *outcome indistinguishability* definition(Dwork et al., 2023):

$$\mathbb{E}_{(\boldsymbol{x},o^*_{\boldsymbol{x}})\sim\mathcal{D}}[A(\boldsymbol{x},\tilde{o}_{\boldsymbol{x}},\tilde{\boldsymbol{p}}) - A(\boldsymbol{x},o^*_x,\tilde{\boldsymbol{p}})] \leq \epsilon, \quad \forall A \in \mathcal{A}$$

where $\mathcal{A}$ is the set of discriminators, $\tilde{o}_{\boldsymbol{x}}$ is the output distribution to learn and $o^*_{\boldsymbol{x}}$ is the true underlying distribution. The goal is to find $\tilde{o}_{\boldsymbol{x}}$ such that all discriminators fail to identify it from the true $o^*_{\boldsymbol{x}}$.

Also, in the context of conformal risk control, there exist two definitions to guarantee *multivalid* coverage, which are also related to our work. The goal in this context is to find a threshold function such that it covers the label for an approximate ratio q. The first definition, denoted as *Threshold Calibrated Multivalid Coverage*, is defined for sequential data (Gupta et al., 2022; Bastani et al., 2022).

Suppose that there are T rounds of data in total, namely $\{(\boldsymbol{x}^{(t)}, y^{(t)})\}_{t=1}^T$. Define $\mathcal{T} \subseteq \mathcal{Y}$ to be the conformal prediction and $s^{(t)}: \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ to be the given scoring function. Denote $q^{(t)}$ as round-dependent threshold, which gives us a prediction set $\mathcal{T}^{(t)} = \left\{y \in \mathcal{Y} : s^{(t)}\left(\boldsymbol{x}^{(t)}, y\right) \leq q^{(t)}\right\}$. Fix a coverage target $(1-\delta)$ and a collection of groups $\mathcal{G} \subset 2^{\mathcal{X}}$.

A sequence of conformity thresholds $\{q^{(t)}\}_{t=1}^T$ is said to be (θ, m)-*multivalid* (Jung et al., 2022) with respect to δ and $\mathcal{G}$ for some function $\theta : \mathbb{N} \to \mathbb{R}$, if for every $i \in [m] \triangleq \{1, 2, ..., m\}$ and $G \in \mathcal{G}$, the following holds true:

$$\left|\bar{H}\left(G^{(T)}(i)\right) - (1-\delta)\right| \leq \theta\left(\left|G^{(T)}(i)\right|\right).$$

where

$$G^{(t)}(i) = \left\{\tau \in [t] : x_\tau \in G, q^{(t)} \in [\frac{i-1}{m}, \frac{i}{m})\right\},$$

$$\bar{H}(S) = \frac{1}{|S|}\sum_{t\in S} \mathbb{1}_{\{s^{(t)} \leq q^{(t)}\}}).$$

The second definition, denoted as *q-quantile predictor* is defined for batch data (Jung et al., 2022). Using $g'(\boldsymbol{x}) = 1$ to denote the membership of certain subgroup of x, the quantile calibration error of q-quantile predictor $f : \mathcal{X} \to [0, 1]$ on group g' is:

$$Q(f, g') = \sum_{v \in R(f)} \mathbb{P}_{(\boldsymbol{x},s)\sim\mathcal{S}}(f(\boldsymbol{x}) = v \mid g'(\boldsymbol{x}) = 1)\left(q - \mathbb{P}_{(\boldsymbol{x},s)\sim\mathcal{S}}(s \leq f(\boldsymbol{x}) \mid f(\boldsymbol{x}) = v, g'(\boldsymbol{x}) = 1)\right)^2.$$

We say that f is α-approximately q-quantile multicalibrated with respect to group collection $\mathcal{G}'$ if

$$Q(f, g') \leq \frac{\alpha}{\mathbb{P}_{(\boldsymbol{x},s)\sim\mathcal{S}}(g'(\boldsymbol{x}) = 1)} \quad \text{for every } g' \in \mathcal{G}'.$$

E.2. Expressing These Definitions as Generalized Multicalibration

We prove that the definition can be reduced to both the definition of HappyMap and the definition of Indistinguishability.

E.2.1. s-HappyMap

Recall that the definition of s-HappyMap is

$$\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[c(f(\boldsymbol{x}),\boldsymbol{x})s(f(\boldsymbol{x}),y)] \leq \alpha, \quad \forall c \in \mathcal{C}.$$

And $(\boldsymbol{s}, \mathcal{G}, \alpha)-$GMC (where $\boldsymbol{s}(\boldsymbol{x}) \in \mathbb{R}^m$) is defined as:

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\langle \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}))\boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D})\rangle] \leq \alpha, \quad \forall \boldsymbol{g} \in \mathcal{G}.$$

We transform $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC into s-HappyMap by defining the following function: (The left side of the equation represents the notation of $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC, while the right side represents the notation of s-HappyMap.)

$$m = 1, \mathcal{G} = \mathcal{C}, g = c, f(\boldsymbol{x}) \in \mathbb{R}, s(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}) = s(f(\boldsymbol{x}), y).$$

Then we have

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\langle s(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), g(f(\boldsymbol{x}), \boldsymbol{x})\rangle] = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[s(f(\boldsymbol{x}), y)c(f(\boldsymbol{x}), \boldsymbol{x})].$$

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},\boldsymbol{y})\sim\mathcal{D}}[\langle s(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D}), g(f(\boldsymbol{x}), \boldsymbol{x})\rangle] \le \alpha, \forall g \in \mathcal{G}.$$

$$\Leftrightarrow \quad \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}[s(f(\boldsymbol{x}), y)c(f(\boldsymbol{x}), \boldsymbol{x})] \le \alpha, \forall c \in \mathcal{C}.$$

Thus, the formulation of $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC is reduced to the s-HappyMap.

E.2.2. Outcome Indistinguishability

Recall the definition of the outcome indistinguishability is

$$\mathbb{E}_{(\boldsymbol{x},o^*_{\boldsymbol{x}})\sim\mathcal{D}}[A(\boldsymbol{x}, \tilde{o}_{\boldsymbol{x}}, \tilde{\boldsymbol{p}}) - A(\boldsymbol{x}, o^*_x, \tilde{\boldsymbol{p}})] \le \epsilon, \quad \forall A \in \mathcal{A}$$

where $\mathcal{A}$ is the set of discriminators, $\tilde{o}_{\boldsymbol{x}}$ is the output distribution to learn and $o^*_{\boldsymbol{x}}$ is the true underlying distribution. The goal is to find $\tilde{o}_{\boldsymbol{x}}$ such that all discriminators fail to identify it from the true $o^*_{\boldsymbol{x}}$.

And $(\boldsymbol{s}, \mathcal{G}, \alpha)-$GMC (where $\boldsymbol{s}(\boldsymbol{x}) \in \mathbb{R}^m$) is defined as:

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},y)\sim\mathcal{D}}[\langle \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{x}))\boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, \boldsymbol{h}, \boldsymbol{y}, \mathcal{D})\rangle] \le \alpha, \quad \forall \boldsymbol{g} \in \mathcal{G}.$$

We transform $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC into Outcome Indistinguishability by defining the following function: (The left side of the equation represents the notation of $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC, while the right side represents the notation of Outcome Indistinguishability.)

$O = \Delta O = \{x \in [0,1]^m : \sum_{i=1}^m x_i = 1\}, \quad \boldsymbol{f}(\boldsymbol{x}) = \tilde{\boldsymbol{p}}(\boldsymbol{x}) \in \mathbb{R}^m$ is the probability distribution of over the output space. Denote $\mathcal{Y} = \{\mathcal{Y}_1, \mathcal{Y}_2, ..., \mathcal{Y}_K\}$ and $\boldsymbol{p}^*_y = (\mathbb{P}[y = \mathcal{Y}_1], \mathbb{P}[y = \mathcal{Y}_2], ..., \mathbb{P}[y = \mathcal{Y}_K])$, we further set

$$\boldsymbol{s}(\tilde{\boldsymbol{p}}, \boldsymbol{x}, h, y, \mathcal{D}) = \tilde{\boldsymbol{p}}(\boldsymbol{x}) - \boldsymbol{p}^*_y \in \mathbb{R}^m. \quad \mathcal{G} = \{\boldsymbol{g}(\tilde{\boldsymbol{p}}(\boldsymbol{x}), \boldsymbol{x}) = A(\boldsymbol{x}, \cdot, \tilde{\boldsymbol{p}}) \in \mathbb{R}^m : A \in \mathcal{A}\}.$$

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},y)\sim\mathcal{D}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, h, y, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle] = \mathbb{E}[A(\boldsymbol{x}, \tilde{o}_{\boldsymbol{x}}, \tilde{\boldsymbol{p}}) - A(\boldsymbol{x}, o^*_x, \tilde{\boldsymbol{p}})].$$

$$\mathbb{E}_{(\boldsymbol{x},\boldsymbol{h},y)\sim\mathcal{D}}[\langle \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}, h, y, \mathcal{D}), \boldsymbol{g}(\boldsymbol{f}(\boldsymbol{x}), \boldsymbol{x})\rangle] \le \alpha, \forall \boldsymbol{g} \in \mathcal{G}. \quad \Leftrightarrow \quad \mathbb{E}[A(\boldsymbol{x}, \tilde{o}_{\boldsymbol{x}}, \tilde{\boldsymbol{p}}) - A(\boldsymbol{x}, o^*_x, \tilde{\boldsymbol{p}})] \le \alpha, \forall A \in \mathcal{A}.$$

Thus, the formulation of $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC is reduced to the Outcome Indistinguishability.

E.2.3. Multivalid Prediction

Recall the definition of the $(\theta, m)-$multivalid prediction is

$$\left|\bar{H}\left(G^{(T)}(i)\right) - (1-\delta)\right| \le \theta\left(\left|G^{(T)}(i)\right|\right). \quad \forall G \in \mathcal{G}, \forall i \in [m].$$

where

$$G^{(t)}(i) = \left\{\tau \in [t] : x_\tau \in G, q^{(t)} \in [\frac{i-1}{m}, \frac{i}{m})\right\},$$

$$\bar{H}(S) = \frac{1}{|S|}\sum_{t\in S} \mathbb{1}_{\{s^{(t)} \le q^{(t)}\}}.$$

Recall that the finite-sampling version $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC (where $\boldsymbol{s}(\boldsymbol{x}) \in \mathbb{R}^m$) is defined as:

$$\frac{1}{T}\sum_{i=1}^{T}[\langle \boldsymbol{g}(\boldsymbol{x}_{(i)}, \boldsymbol{f}(\boldsymbol{x}_{(i)})), \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}_{(i)}, \boldsymbol{h}, \boldsymbol{y}_{(i)}, \mathcal{D})\rangle] \le \alpha, \quad \forall \boldsymbol{g} \in \mathcal{G}.$$

Where T denotes the sample size, $\{(\boldsymbol{x}_{(i)}, \boldsymbol{y}_{(i)})\}$ denotes the data. We transform the finite sampling version of $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC into $(\theta, m)-$multivalid by defining the following function: (The left side of the equation represents the notation of $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC, while the right side represents the notation of $(\theta, m)-$multivalid.)

Let $m = 1, h = s, f = q, s(f, \boldsymbol{x}, h, y, \mathcal{D}) = \mathbb{1}_{\{s \leq q\}} - (1-\delta), \mathcal{G} = \{\mathbb{1}_{\{x \in G\}} \mathbb{1}_{q \in [\frac{i-1}{m}, \frac{i}{m})} : G \in \mathcal{G}, i \in [m]\}, \theta(|G^{(t)}(i)|) = \frac{\alpha T}{|G^{(t)}(i)|}$.

When $g(f(\boldsymbol{x}), \boldsymbol{x}) = \mathbb{1}_{\{x \in G\}} \mathbb{1}_{\{q \in [\frac{i-1}{m}, \frac{i}{m})\}}$,

$$
\begin{aligned}
\frac{1}{T} \sum_{i=1}^{(T)} [\langle \boldsymbol{g}(\boldsymbol{x}_{(i)}, \boldsymbol{f}(\boldsymbol{x}_{(i)})) \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}_{(i)}, \boldsymbol{h}, \boldsymbol{y}_{(i)}, \mathcal{D}) \rangle] &= \frac{1}{T} \sum_{t=1}^{T} g(q(\boldsymbol{x}^{(t)}), \boldsymbol{x}^{(t)}) (\mathbb{1}_{\{s(\boldsymbol{x}^{(t)}, y^{(t)}) \leq q(\boldsymbol{x}^{(t)})\}} - (1-\delta)) \\
&= \frac{|G^{(T)}(i)|}{T} |\bar{H}(G^{(T)}(i)) - (1-\delta)| \\
&\leq \alpha \\
\Leftrightarrow |\bar{H}\left(G^{(T)}(i)\right) - (1-\delta)| \leq \frac{\alpha T}{|G^{(T)}(i)|} &= \theta(|G^{(T)}(i)|)).
\end{aligned}
$$

So

$$
\begin{aligned}
&\frac{1}{T} \sum_{i=1}^{T} [\langle \boldsymbol{g}(\boldsymbol{x}_{(i)}, \boldsymbol{f}(\boldsymbol{x}_{(i)})) \boldsymbol{s}(\boldsymbol{f}, \boldsymbol{x}_{(i)}, \boldsymbol{h}, \boldsymbol{y}_{(i)}, \mathcal{D}) \rangle] \leq \alpha, \quad \forall \boldsymbol{g} \in \mathcal{G} \\
\Leftrightarrow \quad &|\bar{H}\left(G^{(T)}(i)\right) - (1-\delta)| \leq \frac{\alpha T}{|G^{(T)}(i)|} = \theta(|G^{(T)}(i)|)), \forall G \in \mathcal{G}, \forall i \in [m].
\end{aligned}
$$

Thus, the sampling version of $(\boldsymbol{s}, \mathcal{G}, \alpha)-$GMC is reduced to the MultiValid Prediction where θ is of a specific form.

E.2.4. Quantile Multicalibration

The quantile calibration error of q-quantile predictor $f : \mathcal{X} \to [0, 1]$ on group g' is:

$$
Q(f, g') = \sum_{v \in R(f)} \mathbb{P}_{(\boldsymbol{x}, s) \sim \mathcal{S}}(f(\boldsymbol{x}) = v \mid g'(\boldsymbol{x}) = 1) \left(q - \mathbb{P}_{(\boldsymbol{x}, s) \sim \mathcal{S}}(s \leq f(\boldsymbol{x}) \mid f(\boldsymbol{x}) = v, g'(\boldsymbol{x}) = 1)\right)^2 .
$$

Recall that f is α-approximately q-quantile multicalibrated with respect to group collection $\mathcal{G}'$ if

$$
Q(f, g') \leq \frac{\alpha}{\mathbb{P}_{(\boldsymbol{x}, s) \sim \mathcal{S}}(g'(\boldsymbol{x}) = 1)} \quad \text{for every } g' \in \mathcal{G}'.
$$

where $g'(\boldsymbol{x}) = 1$ denotes that x belong to a certain subgroup. This is equal to

$$
q^2 \mathbb{E}_{(\boldsymbol{x}, s) \sim \mathcal{S}}[\mathbb{1}_{\{g'(\boldsymbol{x})=1\}}] - 2q \mathbb{E}_{(\boldsymbol{x}, s) \sim \mathcal{S}}[\mathbb{1}_{\{s \leq f\}} \mathbb{1}_{\{g'(\boldsymbol{x})=1\}}] + \sum_{v \in R(f)} \frac{(\mathbb{E}_{(\boldsymbol{x}, s) \sim \mathcal{S}}[\mathbb{1}_{\{s \leq f\}} \mathbb{1}_{\{g'(\boldsymbol{x})=1\}} \mathbb{1}^2_{\{f(\boldsymbol{x})=v\}})]}{\mathbb{E}_{(\boldsymbol{x}, s) \sim \mathcal{S}}[\mathbb{1}_{\{f(\boldsymbol{x})=v\}} \mathbb{1}_{\{g'(\boldsymbol{x})=1\}}]} \leq \alpha,
$$

$$
\forall g' \in \mathcal{G}'.
$$

We set $\mathcal{G} = \{\pm \mathbb{1}_{\{g'(\boldsymbol{x})=1\}} : g' \in \mathcal{G}'\}, s(f, \boldsymbol{x}, h, y, \mathcal{D}) = q - \mathbb{1}_{\{s \leq f\}}$, then $\mathbb{E}_{(\boldsymbol{x}, h, y) \sim \mathcal{D}} = \mathbb{E}_{(\boldsymbol{x}, s) \sim \mathcal{S}}[g(\boldsymbol{x})(q - \mathbb{1}_{\{s \leq f\}})]$. (The left side of the equation represents the notation of $(\boldsymbol{s}, \mathcal{G}, \alpha)$-GMC, while the right side represents the notation of α-approximately q-quantile.) We have that

$$
|\mathbb{E}_{(\boldsymbol{x}, s) \sim \mathcal{S}}[\mathbb{1}_{\{g'(\boldsymbol{x})=1\}}(q - \mathbb{1}_{\{s \leq f\}})]| \leq \alpha \quad \forall g' \in \mathcal{G}'.
$$

This is equal to

$$
|q \mathbb{E}_{(\boldsymbol{x}, s) \sim \mathcal{S}}[\mathbb{1}_{\{g'(\boldsymbol{x})=1\}}] - \mathbb{E}_{(\boldsymbol{x}, s) \sim \mathcal{S}}[\mathbb{1}_{\{g'(\boldsymbol{x})=1\}} \mathbb{1}_{\{s \leq f\}}]| \leq \alpha \quad \forall g' \in \mathcal{G}'.
$$

The two definitions can't be reduced to each other because of the presence of the term $\sum_{v \in R(f)} \frac{(\mathbb{E}_{(\boldsymbol{x}, s) \sim \mathcal{S}}[\mathbb{1}_{\{s \leq f\}} \mathbb{1}_{\{g'(\boldsymbol{x})=1\}} \mathbb{1}^2_{\{f(\boldsymbol{x})=v\}})]}{\mathbb{E}_{(\boldsymbol{x}, s) \sim \mathcal{S}}[\mathbb{1}_{\{f(\boldsymbol{x})=v\}} \mathbb{1}_{\{g'(\boldsymbol{x})=1\}}]}$. Despite their differences, they both provide robust measures for assessing the extent of coverage in the given context.