Ecole Polytechnique Fédérale de Lausanne
Professor Lénaïc Chizat

Master's Thesis

Exploring Intersectional Fairness in Histopathological Image Classifications

Gaspard Villa
Master in Applied Mathematics - Minor in CSE

supervised by Professor Kun-Hsing Yu
and Shih-Yen Lin, PhD

Boston - March 11, 2024

Contents

1 Introduction **1**

2 Exploratory data analysis **2**
2.1 Presentation of the datasets . . . 2
2.2 Presentation of the protected attributes and the subgroups present in the different datasets 3
2.3 Data structure . . . 5

3 Intersectional fairness metrics **6**
3.1 Definition of intersectional fairness . . . 6
3.2 Minimax Pareto Fairness . . . 6
3.3 Declinations of the Minimax Pareto Fairness metric . . . 8
3.3.1 MinimaxParetoSum5% and MinimaxParetoSum10% . . . 8
3.3.2 MinimaxParetoSumAdapted . . . 9
3.3.3 MinimaxParetoSize . . . 9
3.4 Empirical verification of the metrics . . . 9
3.4.1 Generate different scenarios . . . 10
3.4.2 Comparison of the stability and the accuracy of the metrics . . . 12

4 Construction of the Model and the training technique to ensure intersectional fairness **13**
4.1 Baseline model and method . . . 13
4.1.1 Presentation of the architecture of the model . . . 13
4.1.2 Presentation of the training method . . . 14
4.1.3 Performances of the model on the different datasets . . . 15
4.2 Empirical Differential Fairness method . . . 16
4.2.1 Presentation of the training method . . . 17
4.2.2 Different approaches for the penalty term's computation . . . 18
4.2.3 Selection of the hyperparameters of the method regarding its performances . . . 18
4.3 Two players game formulation of the intersectional fairness problem - MinimaxFair . . . 19
4.3.1 Presentation of the training method . . . 19
4.3.2 Performances of the model on the different datasets . . . 20
4.4 Approximate Projection onto Star sets - APStar . . . 21
4.4.1 Presentation of the training method . . . 21
4.4.2 Performances of the model on the different datasets . . . 23

5 Comparison of the performances of the different methods **24**

6 Remove the protected attributes from input to the model **26**
6.1 Introduction . . . 26
6.2 Performances . . . 26

7 Discussion **29**

8 Conclusion **30**

Abstract

This report explores different training methods to address potential intersectional fairness issues for classification tasks on histopathological images, an area that has received limited attention in prior research. To compare the intersectional fairness performances of different models, four declinations of Minimax Pareto Fairness metrics are introduced, with the $\text{MMPF}_{\text{size}}$ metric chosen as the final comparison metric. Limitations of this metric were identified, including its sensitivity to model uncertainty of its raw predictions and the exclusion of small subgroups inside the data set. Then three training methods were evaluated against a baseline, showing comparable performances and limited improvements at the intersectional fairness level. Potential limitations in the feature extraction process were discussed concerning the final interpretation of the results.

1 Introduction

In recent years, the increasing implementation of machine learning algorithms in different domains generates potential concerns about the fairness of these models. Indeed, some of these models may inadvertently cause harm by giving predictions of unequal accuracy across various subgroups defined by one or multiple protected attributes. While many bias mitigation methods already exist, most of them focus on the marginal definition of fairness: subgroups are solely defined by one protected attributes. And only a few address the more complex field of intersectional fairness, which ensures the fair treatment of individuals belonging to sensitive subgroups defined by multiple protected attributes. As such, ensuring intersectional fairness in the medical field has become a critical area of focus.

Histopathological whole-slide images classification tasks, which involve the automated analysis of tissue samples to detect diseases such as cancer, is one of the field where intersectional fairness can represent a challenge. These tasks often involve complex datasets with diverse subgroups defined by patient demographics. It is then essential to develop methods ensuring equitable predictions across the different subgroups.

To measure this intersectional fairness bias, researchers have proposed multiples fairness metrics. Among these, the MiniMax Pareto Fairness (MMPF) metric is often used to evaluate potential bias in models predictions at a sensitive subgroup level. Instead of aiming to have similar performances across different demographic subgroups, which could lead to unnecessary harm to best-performing subgroups, this metric seeks to minimize the model's error on the worst-performing subgroups. However, the application of the MMPF metric to the specific histopathological image classification tasks we have, is not without its challenges. Indeed, the unequal distribution of data among subgroups presents a significant challenge for this metric. Therefore, we introduce and evaluate four variations of the MMPF metric to address this challenge.

We also aim through this report to evaluate three existing training methods in the context of histopathological image classification, designed to address the intersectional fairness issue. Using one of the variations of the MMPF metric, we compare all these methods against a baseline approach.

2 Exploratory data analysis

2.1 Presentation of the datasets

Before diving into the variations of the MMPF metric and the different training methods explored in this report, we introduce the different datasets and their associated tasks. As outlined in the introduction chapter, our focus revolves around histopathological whole-slide images. These whole-slide images pose a challenge due to their size, rendering a deep learning approach without huge pre-processing steps almost completely ineffective. To address this, we employ the Clinical Histopathology Imaging Evaluation Foundation (CHIEF) [1] model, still under review for publication, to extract multiple features from each slice extracted from the whole-slide images. To briefly introduce this model, the training method uses histopathological images and text information from the pathology report as input. It establishes a histopathological image branch for image encoding (using self-supervised CTransPath model [2]) and another text branch for anatomic site encoding (using the pre-trained text encoder from the CLIP model [3]). This large model was trained with $60,530$ whole-slide images from eight large study consortia (TCGA, GTex, PAIP, PANDA, Basal Cell Carcinomas (BCC), Early Breast Cancer Core-Needle Biopsy (BCNB), AutomatiC Registration Of Breast cAncer Tissue (ACROBAT) and Treatment effectiveness to Ovarian Cancer (TOC)) and five institutional datasets (JUN-QDU-Breast, JUN-QDU-Eso, JUN-QDU-Sto, SYSU-SZ-Cervix, and SYSU–SZ-Endo).

For the purpose of our experiments, we use the labeled histopathological whole-slide images from the TCGA [4] dataset. These images are distributed among various tasks to achieve slide-level classification for each patient. Consequently, two distinct classification tasks are used to demonstrate the performances of the different methods presented in this report: firstly, the detection of different cancers (one cancer type at a time), with a total of 8 cancer types considered. Secondly, an expected more complex task implies cancer classification, where the objective is to identify one cancer type from two possibilities for each patient (8 tasks). With these 8 cancer classification tasks, we get a total of 16 different classification tasks. The following Table 1 summarizes the different studied cancers and their corresponding abbreviations used through the report to simplify the notations.

Table 1 – All the cancers studied through this report with their corresponding abbreviation.

Study abbreviation	Study name	Study abbreviation	Study name
brca	Breast invasive carcinoma	kirp	Kidney renal papillary cell carcinoma
coad	Colon adenocarcinoma	luad	Lung adenocarcinoma
kich	Kidney Chromophobe	lusc	Lung squamous cell carcinoma
kirc	Kidney renal clear cell carcinoma	read	Rectum adenocarcinoma

For cancer classification tasks, we differentiate two types of samples: one is for frozen sections (abbreviated FS later), and the second one is for permanent slides (abbreviated PM) or more precisely Formalin-Fixed Paraffin-Embedded (FFPE) tissues. Then to designate the cancer classification task to classify Lung adenocarcinoma (luad) cancer and Lung squamous cell carcinoma (lusc) cancer for frozen sections (FS) samples, we use the notation: luad_lusc_FS. On the histogram presented in Figure 1, we can observe the number of patients for each classification task. A lot of differences may be seen between each task, meaning to stay aware of this fact for a later interpretation of the results. Indeed, the classification tasks with a few amount of data are expected to be harder to solve.

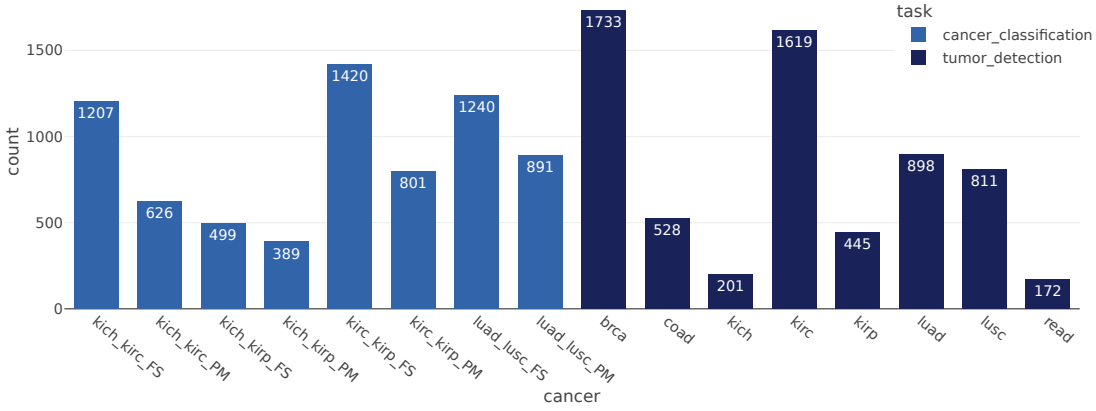


Figure 1 – Number of patients for each experiment depending of the classification tasks and the cancer(s) we are trying to identify.

2.2 Presentation of the protected attributes and the subgroups present in the different datasets

The classification tasks we are working with may present some intersectional fairness issue we try to solve through various methods presented later in this report. To define the subgroups on which a fairness criterion is required to be respected, we are using three protected attributes attached to every patient: the age, the race and the gender.

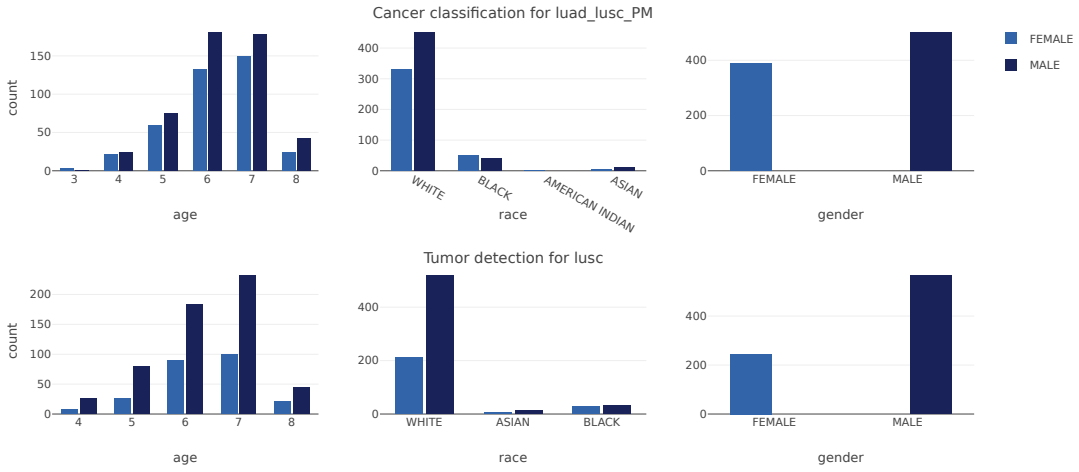


Figure 2 – Number of patients for one cancer classification task (luad_lusc_PM) and one tumor detection task (lusc) on the marginal subgroups formed by each of the three single protected attributes: age, race and gender.

The age is defined as an integer *age* between 0 and 9, representing the range of the patient's age defined by the following interval: $[10 \times age; (10 \times (age + 1)) - 1]$. In Figure 2, the age distribution of the patients from two classification tasks (cancer classification for luad_lusc_PM and tumor detection for lusc) is shown in histograms on the left side of the figure. We observe that patients from both datasets (as well as the remaining tasks) are sparsely distributed between 40 and 80 years old. Races are classified into four different categories: blacks, asian, american indian and white. Their distribution among the patients is depicted in histograms placed at the center of Figure 2. We clearly observe that some races are over-represented (c.f. white) compared to others (c.f. black, american indian or asian). Finally, genders are categorized between male and female, with their distribution shown in histograms on the right side of Figure 2. The male category appears to be generally represented twice more than the female category.

With the three protected attributes being presented, we can now discuss the different subgroups we are facing through the various classification tasks. Theoretically, we would expect to work with $10 \times 4 \times 2 = 80$ subgroups if each category of the protected attributes were evenly represented. However, as mentioned earlier, the age distribution is not evenly spread across the 10 categories. Because of that, we more frequently encounter around 20 to 40 different subgroups. As illustrated in Figure 3, we observe a sparse distribution of patients among these intersectional subgroups.

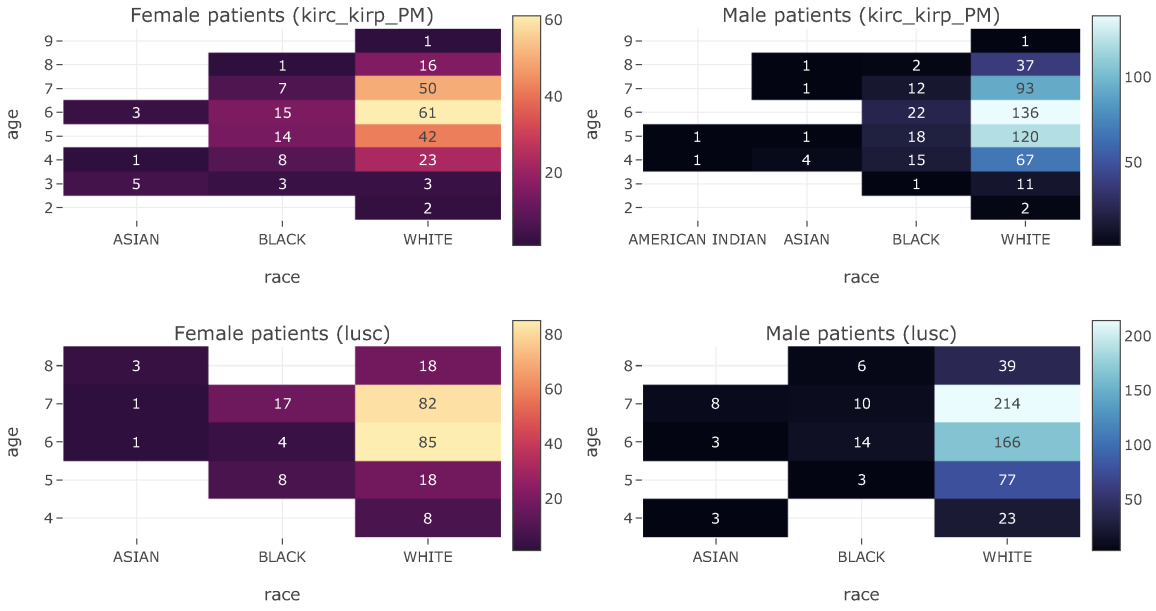


Figure 3 – Density maps of the number of patients for one cancer classification task (kirc_kirp_PM) and one tumor detection task (lusc) in the intersectional subgroups formed by each of the three single protected attributes: age, race and gender.

In some subgroups, the number of patients may be as low as 1, presenting an interpretability issue for the analysis of the results later on. This concern will be addressed in Section 3 concerning the metrics used to compare different models.

2.3 Data structure

The different tasks being presented, we can move on the more technical aspects of the data. Each slide (associated with one patient) is sliced into N_{slices} slices of constant size of 256×256 pixels, from which we extract N_{features} features. This results in tensors of size $N_{\text{slices}} \times N_{\text{features}}$ for each patient. Specifically, we have a constant number of features $N_{\text{features}} = 768$, while the number of slices N_{slices} may vary due the different sizes of each slide images among patients and the constant size of slices.

This particularity presents challenges for the training phase, as batch processing during this phase requires tensors of constant size. To address this issue, we chose to use a random sampling with replacement for the slices of each patient selected for the training set, ensuring a constant tensor size of $\tilde{N}_{\text{slices}} \times N_{\text{features}}$. This method selects randomly $\tilde{N}_{\text{slices}} = 200$ slices among the N_{slices} available for each patient, with a possibility for slices to be selected multiple times. This idea of random sampling with replacement is necessary because some whole-slide images are too small to yield $\tilde{N}_{\text{slices}} = 200$ unique slices. However, reducing $\tilde{N}_{\text{slices}}$ would penalize the training process, so it represents a trade-off between having duplicates and ensuring sufficient information for the training process.

In addition to these tensors, we also decided to give the protected attributes information of each patient to the model as input. The age, race and gender of the patient are then categorized using integers indices and added at the end of the data tensor up scaling its size to $\tilde{N}_{\text{slices}} \times (N_{\text{features}} + 3)$. The full process described before is explained in the following Diagram 4.

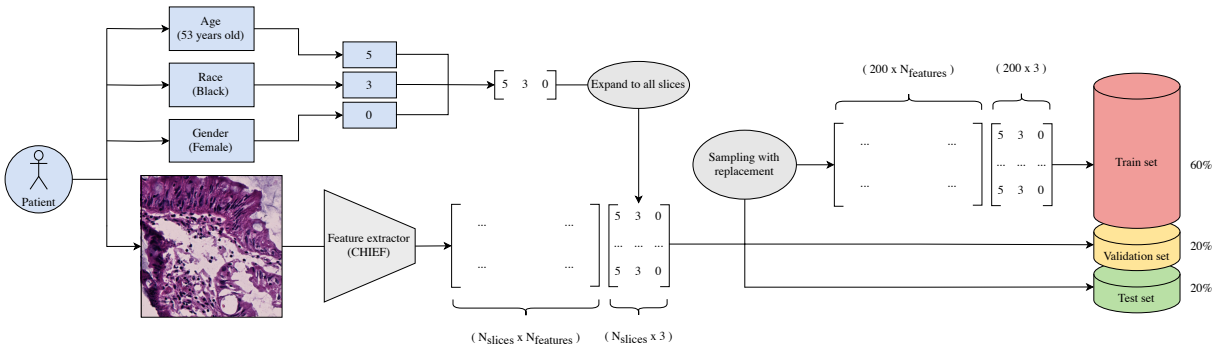


Figure 4 – Simplified diagram of the data processing procedure, from raw data to distribution between training, validation and test sets.

One precision about the splitting process of the data between training, validation and testing set, this is a classic split using the ratio distribution (60% / 20% / 20%) between respectively train, validation and test. But because of the data sparsity across the subgroups defined by the protected attributes (observable in Figure 2), we also apply this ratio distribution (60% / 20% / 20%) at the subgroup level to maintain this data distribution among the training, validation or testing subsets. For the subgroups with less than 3 samples in it, we prioritize on the training set first, then the validation and finally the testing.

Furthermore, the distribution of "positive" (1) or "negative" (0) labels is in the same maintained through the train, validation and test sets. The notion of "positive" or "negative" label is unclear for cancer classification and will be discussed later in Section 4.1.3.

3 Intersectional fairness metrics

3.1 Definition of intersectional fairness

As presented in the introduction of this report, the precise definition of intersectional fairness is challenging to pose. To address this, we rely on key concepts from existing research and make adjustments to some metrics proposed for measuring the fairness behaviour of some predictions. Many studies, such as [5], [6], and [7], have highlighted the bias inherent in AI models. It shows the importance of careful tracking of fairness and bias issues when developing such models. A common observation of a fairness or bias issue happens when model accuracy is inconsistent regarding one or many protected attributes (such as age, race or gender as in the scope of this report). Many methods already exist to ensure marginal fairness at the single protected attributes level, but achieving fairness regarding multiple protected attributes (i.e. intersectional fairness) is a more challenging task.

The diagram presented in Figure 5 highlights the simple work case where predictions given by a model may be fair regarding single protected attributes and still be unfair across the multiple attributes. Indeed, the accuracy of the predictions is consistent across the single protected attributes. But the predictions at the intersectional subgroup level are clearly sparse, indicating unfairness regarding multiple protected attributes.

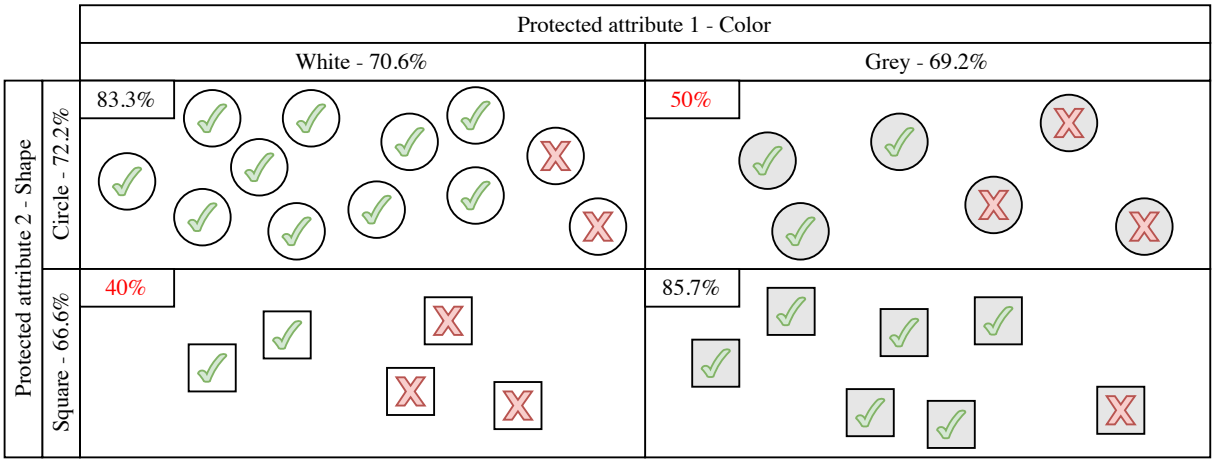


Figure 5 – Illustrative example of how marginal fairness does not ensure intersectional fairness. Fairness seems satisfied regarding the subgroups formed by single protected attributes (color or shape), but we have unfairness for subgroups formed by multiple protected attributes (color and shape).

3.2 Minimax Pareto Fairness

To face off intersectional fairness, we need new metrics to describe it as classic scores are obsolete for this task. The idea of these new metrics is to show whether predictions are consistent regarding the subgroups. One logical approach could be using the variance of the accuracy between the different subgroups and trying to minimize this variance. Indeed, the consistency of the predictions is highly encouraged using this kind of metric. But focusing our training on optimizing this metric may result in under-performances on the best-performing subgroups in order to be consistent with the worst-performing ones. This would cause unnecessary harm to some subgroups of the data to rely on the proposed variance metric.

To face the intersectional fairness issue without introducing unnecessary harm, an approach given by [8] is to use the Minimax Pareto Fairness metric. The idea is simple: instead of trying to minimize the variance of accuracy between the subgroups, we focus on the worst performing subgroup and try to improve the model on it. This method avoids deliberately or inadvertently under-performing the model on the best subgroups and instead encourages the model to improve its predictions on the worst-performing subgroups.

To formalize this notion, we have to define some notations. We consider a trainable classifier $h \in \mathcal{H}$ from a hypothesis class $\mathcal{H}$. The set of features on which this classifier bases its predictions is noted $X = \{x_i\}_{i=1,\dots,N}$ and the corresponding response variables rely in the set $Y = \{y_i\}_{i=1,\dots,N}$, where N is the total number of data in a single dataset. The set of subgroups defined by the protected attributes of each sample is noted $A = \{a_i\}_{i=1,\dots,N_A}$, where N_A is the number of subgroups, each of size N_{a_i}. We notice the features defining whether the sample belongs to a specific subgroup may or may not be in the features set X. For now, we consider the case where these protected features are inside the features space X as specified in the diagram from Figure 4. To train the classifier h on the dataset $\mathcal{D} = X \times Y$, we define a loss function as follow:

$$\mathcal{L} : \begin{array}{c} Y \times Y \to \mathbb{R} \\ (\tilde{y}, y) \to \mathcal{L}\left(\tilde{y}, y\right) \end{array} \tag{1}$$

where $h(x) = \tilde{y}$ is the prediction given by the classifier h on features x. To simplify the notations, we introduce the notion of risk given a classifier h on a subgroup $a \in A$, noted $r_h(a)$. The risk is defined as the average loss evaluated on the predictions given by h from the data belonging to the specific subgroup a. It is expressed as follow:

$$r_h(a) = \frac{1}{N_a} \sum_{(x,y)\in a} \mathcal{L}(h(x), y) \tag{2}$$

We also introduce the risk vector $\mathbf{r}_h \in \mathbb{R}^{N_A}$ for a classifier $h \in \mathcal{H}$ where each of its element represents the risk evaluated on a corresponding subgroup. Formally, it is written as $(\mathbf{r}_h)_i = r_h(a_i)$, for $i = 1, \dots, N_A$. We now present the Minimax Pareto Fairness (MMPF) metric introduced in [8] as follow:

$$\text{MMPF}(h) = \max_{a\in A} r_h(a) = \|\mathbf{r}_h\|_\infty \tag{3}$$

And by introducing this metric, we can formalize the problem of intersectional fairness as a minimization problem of the MMPF metric, which can be expressed as follow:

$$\begin{aligned} \tilde{h} &= \arg\min_{h\in\mathcal{H}} \text{MMPF}(h) \\ &= \arg\min_{h\in\mathcal{H}} \|\mathbf{r}_h\|_\infty \\ &= \arg\min_{h\in\mathcal{H}} \max_{a\in A} \mathcal{L}_{(h,a)} \end{aligned} \tag{4}$$

This interpretation of the intersectional fairness problem is based on the idea to reduce the loss (i.e. the risk error) evaluated on the worst-performing subgroup within the dataset. By minimizing this loss, we improve the performances of predictions from the subgroups that pose the most challenges for the model. Following this idea, we aim to reach a point where the predictions on the worst-performing subgroups can not be improved without penalizing even more the others.

3.3 Declinations of the Minimax Pareto Fairness metric

Before adopting this metric as a benchmark to compare the different methods in terms of intersectional fairness criteria, it is crucial to ensure that it aligns well with our expectations for the specific classification tasks we are working with. The MMPF metric depends directly on the significance of the average loss observed in each subgroup. However, a challenge arises from the sparse representation of some subgroups compared to others, as illustrated in the figure 3. In many cases, some subgroups contains only one patient implying a considerable risk on the interpretation of a such metric. For these scenarios, we could encounter an almost perfect classifier which misclassifies only a single sample within this single-sampled subgroup. The metric would then classifies this model as highly unfair because of the loss of the single misclassified sample. Conversely, we might encounter another model that misclassifies many more samples from larger subgroups but maintains a smaller average loss compared to the unfair model's metric. It would then indicates a model fairer than another when it is not actually the case. This notion of a model fairer than the other is highly open to interpretation, but we can not say that the prediction on a single patient is decisive to class a model unfair when the other is wrong on a larger number of patients. This brings out the necessity of modifying the current MMPF metric to address theses cases involving smaller subgroups.

3.3.1 MinimaxParetoSum5% and MinimaxParetoSum10%

To handle these cases with smaller subgroups, we keep the same idea of the MMPF metric but by only focusing on subgroups of reasonable sizes. For that, we come with the idea of considering the worst-performing subgroups such that their cumulative size reaches a minimum threshold and computing the cumulative average loss on all data from these subgroups. Doing so, we have a stable metric less sensitive to smaller subgroups. It also ensures the general case where the worst-performing subgroup is large enough to be meaningful since it then works as the initial MMPF metric by taking average loss only on this subgroup. It remains now to define the said threshold.

For that, we begin with arbitrary threshold set to 5% of the dataset size N, rounded to the upper integer and noted $N_5 = \lceil 0.05N \rceil$. We note this new metric as MMPF_5. By the definition of N_5, we may not find a sum of samples among the worst-performing subgroups perfectly equal to it. Then, we use the integer N_5 as a minimum threshold to exceed when counting the number of samples from the worst-performing subgroups. To do so, we sort the subgroups regarding their risk error $r_h(a)$ in the descending order and add a sub-sorting (for equal risk situation between at least two subgroups) depending of their size in ascending order to get closer to N_5 as possible. This gives us the following sorted list of subgroups:

$$\bar{A} = \left\{ a_{k_i} \mid r_h(a_{k_i}) > r_h(a_{k_{i+1}}) \text{ or, } r_h(a_{k_i}) = r_h(a_{k_{i+1}}) \text{ and } N_{a_{k_i}} \leq N_{a_{k_{i+1}}} \right\} \tag{5}$$

with the following partition $\{k_i\}_{i=1}^{N_A} \subseteq \{1, \ldots, N_A\}$. We note m_5 the smallest integer such that the sum of sizes of subgroups in $\{a_{k_i}\}_{i=1}^{m_5}$ is larger or equal than N_5. This being said we get the metric MMPF_5 expressed as follow:

$$\text{MMPF}_5 = \frac{1}{m_5} \sum_{i=1}^{m_5} r_h(a_{k_i}) N_{a_{k_i}} \tag{6}$$

In other words, it represents the average loss of all the samples from the set of subgroups $\{a_{k_i}\}_{i=1}^{m_5}$. This is how we define the MMPF_5 metric and the definition for the 10% case follows the same steps by taking 10% instead of 5% of the dataset size and we obtain the following formulation:

$$\text{MMPF}_{10} = \frac{1}{m_{10}} \sum_{i=1}^{m_{10}} r_h(a_{k_i}) N_{a_{k_i}} \tag{7}$$

3.3.2 MinimaxParetoSumAdapted

Using arbitrary threshold such as 5% or 10% lacks significance, we have to think to a more reasonable one. The idea of seeking of a new metric to replace MMPF comes from the sparsity of the data among subgroups. In an ideal scenario, we would have an equal amount of data for each subgroup, implying $\frac{N}{N_A}$ data points per subgroup. Suggesting that the MMPF metric would be more significant when the size of the subgroups getting larger or equal than $\frac{N}{N_A}$. Therefore, we can extend the same concept as in the previous Section 3.3.1 and giving us the $\text{MMPF}_{\text{adapt.}}$ metric defined in (8) where $m_{\text{adapt.}}$ is an integer representing the first $m_{\text{adapt.}}$ subgroups having their sum of sizes at least equal to $\frac{N}{N_A}$.

$$\text{MMPF}_{\text{adapt.}} = \frac{1}{m_{\text{adapt.}}} \sum_{i=1}^{m_{\text{adapt.}}} r_h(a_{k_i}) N_{a_{k_i}} \tag{8}$$

3.3.3 MinimaxParetoSize

Finally, instead of taking all the worst subgroups and compute the average loss on them, we may think to another approach by considering all subgroups under a specific size as non-significant. Following the same concept from previous sections, only subgroups of size larger or equal than $\frac{N}{N_A}$ (the new threshold) are considered and we compute average loss on each of these subgroups.

To build this metric, we consider the subset $\bar{A}_{\text{size}} \subseteq \bar{A}$ that keeps the same sorting of $\bar{A}$ (5) but remove subgroups with size strictly lower than $\frac{N}{N_A}$. More formally we write it $\bar{A}_{\text{size}} = \{a \in \bar{A} \mid N_a \geq \frac{N}{N_A}\}$ and we define the $\text{MMPF}_{\text{size}}$ metric in the following equation (9), where $\bar{a}_0 = \bar{A}_{\text{size}}[0]$ being the first element of $\bar{A}_{\text{size}}$:

$$\text{MMPF}_{\text{size}} = r_h(\bar{a}_0) \tag{9}$$

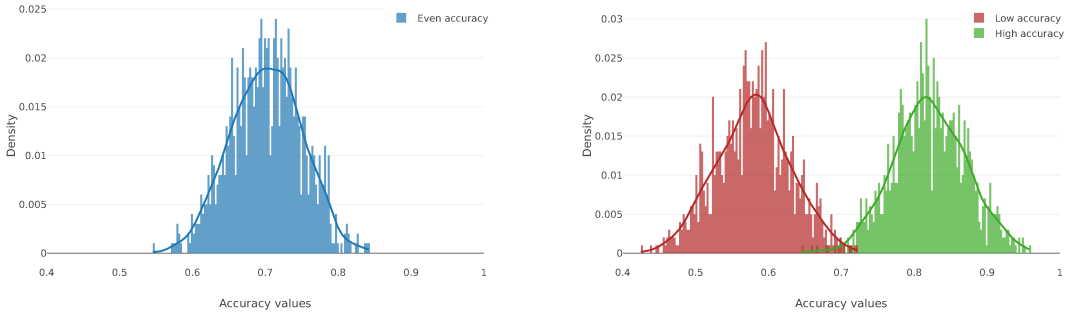
3.4 Empirical verification of the metrics

To choose the most appropriate metric for future tasks, we must assess the accuracy and stability of the various metrics we proposed here. To achieve this, we require a dataset representing a real-world situation and a model to generate predictions from this dataset. One approach could involve using the Baseline model across the different classification tasks we have on histopathological whole-slide images, giving at the same time both the dataset and the model. However, a challenge arises when evaluating the accuracy of the metric. Indeed, an accurate metric has to be consistent on the evaluation it make on fair and unfair prediction scenarios (lower is the value, fairer are the predictions). And assessing whether a scenario is fair or not is subject to interpretation and may be time-consuming to establish manually on the predictions given by the Baseline model. Therefore, another approach is to manually build these datasets and generate synthetic predictions that represent fair or unfair scenarios.

3.4.1 Generate different scenarios

To start on a meaningful basis, we chose not to build our dataset from scratch, but instead use a simplified version of one of the datasets we have from the different classification tasks we have. Instead of considering the three protected attributes *age*, *race* and *gender*, we reduce to two single protected attributes denoted respectively as att_1 and att_2, by ranging both from class values 1 to 6. From the figure 3, we note that for the classification task for the cancers kirc_kirp_PM, there are 37 different subgroups, which we reduce to 36 by excluding one subgroup with only one patient to match the $6 \times 6 = 36$ subgroups of our new dataset. And finally, our new dataset gets in each of its subgroup the same number of data there is in a corresponding subgroup from kirc_kirp_PM, maintaining the same sparsity of the data among subgroups as for kirc_kirp_PM. Following this process, the labels of each corresponding data is respected regarding the ones from kirc_kirp_PM. Consequently, we obtain a dataset characterized by a specific density of data in each subgroup as presented in the left heatmap of Figure 7.

Our dataset being properly defined, we must construct the fair and unfair scenarios on it. We define here the notion of fairness as the ability to achieve comparable predictions accuracy across each subgroup without penalizing one or multiple subgroups compare to the others. Then, a fair scenario is translated as the situation where the performances on each subgroup follows a normal distribution $\mathcal{N}(\mu, \sigma)$. Following the same idea, we define an unfair scenario as a situation where two normal distributions $\mathcal{N}(\mu_1, \sigma_1)$ and $\mathcal{N}(\mu_2, \sigma_2)$ of the subgroups' performances are distinguishable. In realistic situations, there may be more than two distinguishable clusters, but we limit this for the current experiments. Using these definitions, fair and unfair scenarios on our dataset can be randomly generated N_{scenario} times, as shown in Figure 6.



(a) Fair scenario with subgroups accuracy following a normal distribution $\mathcal{N}(0.7, 0.05)$.

(b) Unfair scenario with subgroups accuracy distributed between two normal distributions $\mathcal{N}(0.82, 0.05)$ and $\mathcal{N}(0.58, 0.05)$.

Figure 6 – Distributions of the performances that predictions should satisfy on each subgroup regarding a fair scenario (left) and an unfair one (right) for $N_{\text{scenario}} = 1000$ scenarios.

This random generation process assigns an accuracy rate for each subgroup depending they are in a fair or unfair scenario. Each of these subgroups then generated predictions to reach the accuracy rate associated with them. For a fair scenario, $n = 36$ samples are generated from the distribution $\mathcal{N}(0.7, 0.05)$ and randomly assigned to the 36 subgroups. For the unfair scenario, while 18 samples from the low accuracy distribution (red) are assigned to as many subgroups, 18 other samples from the high accuracy distribution (green) are assigned to the remaining subgroups. The following Figure 7 depicts the accuracy predictions of the subgroups for one random generation of a fair (right) and an unfair scenario (center).

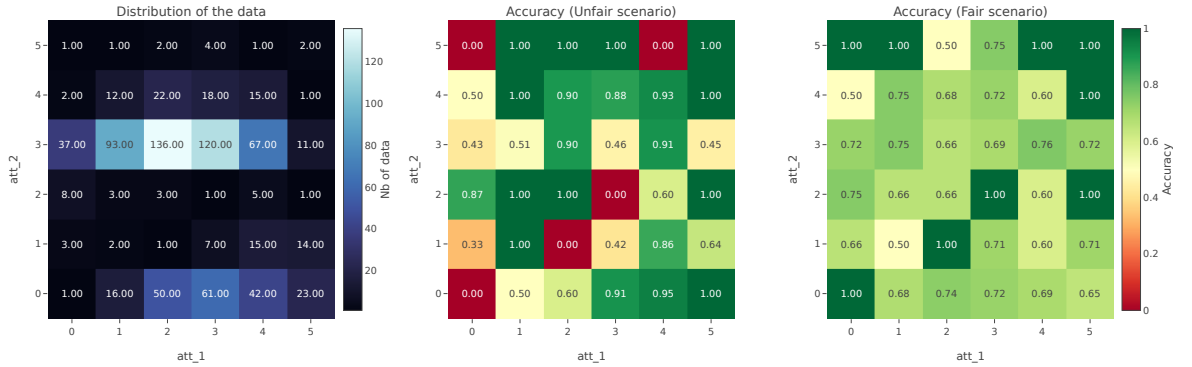


Figure 7 – Distribution of the samples across the subgroups for the new dataset derived from kirc_kirp_PM (left); and examples of subgroups accuracy for an unfair scenario (center) and a fair scenario (right).

All the proposed metrics are based on the average loss value evaluated on each subgroup's predictions. As a loss function, we use the Cross Entropy loss defined later in (11) adapted for binary classification tasks. But to use it, we need the raw predictions from a model (or scenario) represented, for each patient, by a two-dimensional tensor corresponding to the patient's probability belonging to negative class (first element of the tensor) or positive class (second element). To construct these probabilities given by a fair or unfair scenario, we simply use a uniform distribution between 0.5 and 1 for the element in the tensor corresponding to the class determined by the predictions, and between 0 and 0.5 for the other element.

Finally, we use the same splitting method used during the training to generate N_{split} test datasets. These test datasets are the ones on which the various metrics are evaluated to get similar situations we should have with the original dataset kirc_kirp_PM. Then, the accuracy of each metric is evaluated by verifying if its output values correspond to the behavior of a fair and unfair scenario. More precisely, in a fair scenario, the metric value should be smaller than the one evaluated on an unfair scenario. All the procedure is summarized on the diagram from Figure 8.

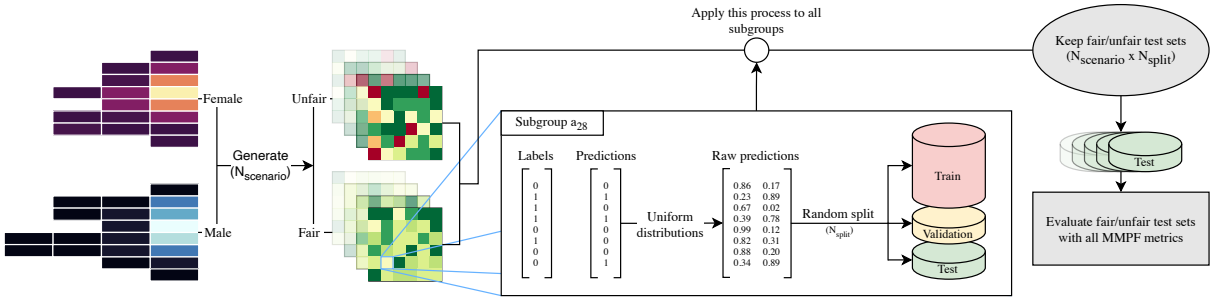
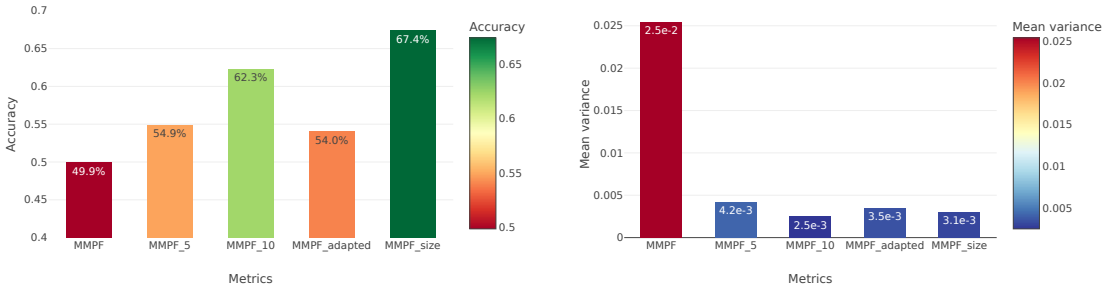


Figure 8 – Diagram of the procedure for the generation of the various fair and unfair scenario to evaluate the different metrics. From a specific dataset (and its subgroup), we generate N_{scenario} fair and unfair scenarios from which N_{split} fair and unfair test sets are generated to finally being evaluated regarding all metrics.

3.4.2 Comparison of the stability and the accuracy of the metrics

Before stepping into the analysis of the general accuracy of each metric, we also need to verify the stability on each method. And by stability, it means the metric need to be consistent independently of the subsets we observe. More precisely, the metric must have a low variance when measuring different test sets on the same classification task. To evaluate this variance, we simply use the predictions given by the Baseline model (presented later, but consider it as an oracle to gets realistic predictions) and measure each metric on a test set obtained using the splitting method presented before. Doing so, we generate a $N_{\text{stability}} = 1000$ random test subsets for each of the 16 different classification tasks on histopathological images. And we evaluate each of these test subsets with the proposed metrics. We then measure the variance of the output values given by every metric on each classification task. Then, the mean of these variances are plotted on Figure 9 (b). For the accuracy, we generate $N_{\text{scenario}} \times N_{\text{split}}$ different cases to evaluate each of our metrics and the metric accuracy are plotted on Figure 9 (a).



(a) Accuracy of each of the five metrics on the handcrafted dataset ($N_{\text{scenario}} = 1000$ and $N_{\text{split}} = 1000$).

(b) Average variance of the evaluation given by each metric on the predictions made by the Baseline model on the given histopathological images classification tasks.

Figure 9 – The accuracy and average variance (stability) for each of the five proposed metrics on $1e6$ handcrafted scenarios for the accuracy and the predictions given by the Baseline model on the 18 original datasets for the stability.

Finally, a clear improvement can be observed for any of the metric we propose compare to the initial one MMPF. For the stability first, any of the four metric we come with are potential good candidates because of their low average variance. It is important for these metrics to have a low variance, ensuring consistent values regardless of the random split applied to generate the test set. We note that the MMPF_{10} and $\text{MMPF}_{\text{size}}$ are the one with the lowest values. For the accuracy, the results are more nuanced. We still have an improvement of our metric compare to the initial MMPF. However, the performances of MMPF_5 and $\text{MMPF}_{\text{adapted}}$ are clearly lowere than those of MMPF_{10} and $\text{MMPF}_{\text{size}}$, where we achieve an accuracy of 64.7% for the latter across $1e6$ different scenarios and test sets. Its accuracy is clearly above the one from MMPF_{10} and they have both a comparable results in terms of consistency. Finally, regarding the presented results, we chose to use $\text{MMPF}_{\text{size}}$ as the metric to evaluate the performances of each model in terms of intersectional fairness.

4 Construction of the Model and the training technique to ensure intersectional fairness

The metric to compare the models being selected, we may present the various methods explored through this report. All these methods focus on the training process that must assess potential intersectional fairness issues. The first model is the Baseline model on which the improvements in terms of intersectional fairness criterion, defined by the evaluation of the predictions regarding the metric $\text{MMPF}_{\text{size}}$, are compared to the others training methods. The following steps are presenting the three various approaches coming from different research papers seemly adapted to the intersectional fairness issue for classification tasks on histopathological whole-slide images. Each approach is formally introduced and after that evaluated regarding its accuracy, F1-score and $\text{MMPF}_{\text{size}}$ values.

4.1 Baseline model and method

4.1.1 Presentation of the architecture of the model

As explained in the exploratory data analysis Section 2, we are not working with the whole-slide histopathological images but with the extracted features from their slices cut from the whole-slide images using the CHIEF method [1]. We present now the Baseline model used to assess the different classification tasks at our disposal (8 tasks for tumor detection and 8 tasks for binary cancer classification).

A widely used method in deep learning for images classification tasks is the attention mechanism presented in [9]. Because of the large number of features extracted from the whole-slide images, we are working in the scenario of Multiple Instances Learning (MIL) introduced in [10] and [11] and later specifically addressed to medical images in [12]. To face this scenario, the architecture is inspired from the gated attention mechanism presented in [13]. It defines a two-layers neural network defining the MIL attention coefficients as learnable parameters. The idea is to associate each of these coefficients as a weight to every slice from whole-slide images sliced by CHIEF method [1]. A diagram corresponding to the architecture of the MIL attention layers is presented in Figure 10.

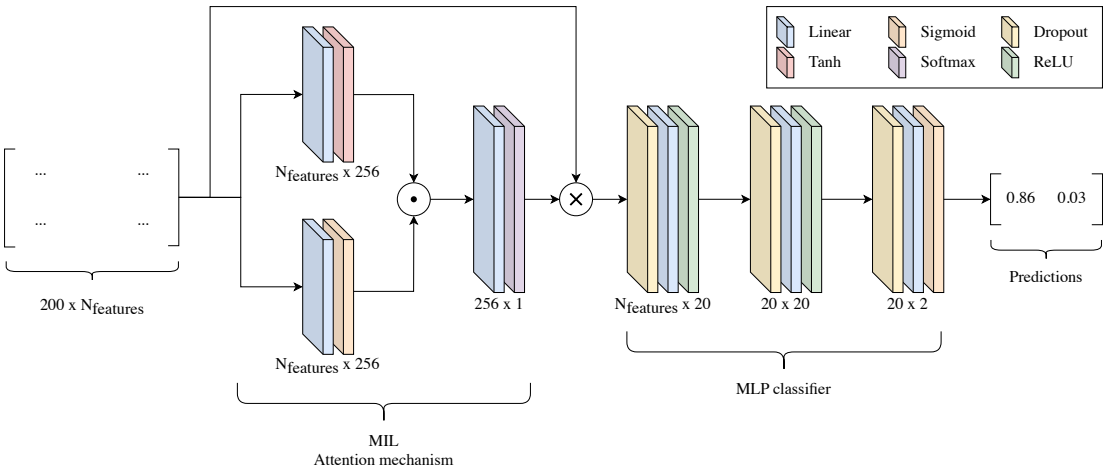


Figure 10 – Diagram representing the Baseline architecture with the MIL attention mechanism for the weights on each slice and the MLP classifier to get the final predictions.

To classify these new features which are each the weighted sum of corresponding features from all slices (one weight for all features of one slice), a simple Multi-Layer Perceptron (MLP) built from three layers is used. The detail of each of these layers is presented in the diagram from Figure 10. The classifier outputs a tensor of length 2 corresponding to the probability prediction of the current sample belonging respectively to the first or second class.

To present in more detail the architecture of this classifier model, the Linear layers are linear transformation of the input $x \in \mathbb{R}^M$ by matrices $A \in \mathbb{R}^{M \times N}$ and $b \in \mathbb{R}^N$ to get an output $y = Ax + b \in \mathbb{R}^N$. There are the main components for Neural Networks models and often preceded with Dropout process. This last process is a regularization method widely used for Deep Learning models presented in [14] as a method to prevent overfitting. It involves ignoring certain neurons during the model training with a predefined probability p (set to 0.5 in our case). The remaining elements used in Figure 10 are element-wise functions to remove the linearity of the computation in a Neural Network model. There are defined as follow:

$$\begin{aligned} \text{Tanh}(x_i) &= \frac{e^{2x_i} - 1}{e^{2x_i} + 1} & \text{Softmax}(x_i) &= \frac{e^{x_i}}{\sum_j e^{x_j}} \\ \text{Sigmoid}(x_i) &= \frac{1}{1 + e^{-x_i}} & \text{ReLU}(x_i) &= \max(0, x_i) \end{aligned} \tag{10}$$

Finally, the two operations presented in the diagram are respectively element-wise multiplication between two matrices and classic matrix product (from left to right). We underscore that the term N_{features} here is equal to $768 + 3 = 771$ because we add the three protected attributes terms as explained in Figure 4. The model we presented in this section is the architecture also used for the future methods. The changes to these are not operated on the architecture of the model, but on the training method we introduce now for Baseline.

4.1.2 Presentation of the training method

The Baseline model is trained using gradient descent method based called *Adam* [15], standing for Adaptive Moment estimation. To present this method, the idea is to update the weights θ, defining the parameters of the model, using exponential moving averages of the gradient of the function $f(\theta)$ (function we want to minimize during training) and the squared gradient. These moving averages represents estimates of the 1st and 2nd moments of the gradient (noted respectively m_k and v_k). The method is characterized by a learning rate α (set to 1e-3 here) and the two exponential decay rates β_1 and β_2 (default equals to 0.9 and 0.999 respectively) for the moment estimates. The general concept of the method is introduced in Algorithm 1.

The value of ϵ, presented in Algorithm 1, is fixed to a small scalar (e.g. 1e-8) to prevent division by zero during the training. The gradient of the function f with respect to the parameters θ are computed using backpropagation method, automatically supported by the `autograd` argument of tensors from PyTorch [16]. It remains to define the loss function which the algorithm *Adam* is minimizing. We chose the Cross Entropy Loss [17] widely used for classification tasks. Because of the 2-dimensional tensor $\tilde{y} = (y_0, y_1)$ raw prediction from our model, the loss is expressed as follow in (11):

$$\mathcal{L}(\tilde{y}, y) = -w_y \log\left(\frac{\exp(\tilde{y}_y)}{\exp(\tilde{y}_0 + \tilde{y}_1)}\right), \tag{11}$$

where the true response is designated as $y \in \{0, 1\}$. As the distribution of the responses 0 and 1 may be very unbalanced across the 16 various classification tasks, we use class weights $w = (w_0, w_1)$ during

the computation of the loss to compensate this difference. The class weights are simply computed as $w_i = \frac{N_0+N_1}{N_i}$ where N_i is the number of samples labelled in true class i, for $i = 0, 1$.

Algorithm 1: ADAM

Input Function to minimize $f(\theta)$ with parameters θ to update;
$m_0 = 0$; $v_0 = 0$;
for $k = 1, \ldots, N_{conv}$ **do**
  Compute gradients: $g_k \leftarrow \nabla_\theta f_k(\theta_{k-1})$;
  Biased 1st moment: $m_k \leftarrow \beta_1 m_{k-1} + (1-\beta_1) g_k$;
  Biased 2nd moment: $v_k \leftarrow \beta_2 v_{k-1} + (1-\beta_2) g_k^2$;
  Unbiased 1st moment estimate: $\hat{m}_k \leftarrow m_k / (1-\beta_1^k)$;
  Unbiased 2nd moment estimate: $\hat{v}_k \leftarrow v_k / (1-\beta_2^k)$;
  Update the parameters of the model: $\theta_k \leftarrow \theta_{k-1} - \alpha \hat{m}_k / (\sqrt{\hat{v}_k} + \epsilon)$;
end
Output Fitted parameters θ_k

4.1.3 Performances of the model on the different datasets

The model being presented, it can be trained and evaluated on the various classification tasks on histopathological whole-slide images. As a reminder, there is two types of tasks: one is for tumor detection of a specific type of cancer, and the other is for cancer classification between two different ones. The training is done using the tools given by PyTorch Lightning [18] to get an efficient optimization for each run. The train loop is 500 epochs long and the best iteration of the model is kept regarding an evaluation with the $\text{MMPF}_{\text{size}}$ metric on the validation set. All the code used to train and compute the results are available in the GitHub repository in [19].

To visualize the performances of the BASELINE method on each task, the accuracy, the F1-score and the introduced $\text{MMPF}_{\text{size}}$ are computed on the predictions given by the model after training and validation on the testing set. The F1-score is a classification metric used to measure the performances of a model on a specific class. Technically, it represents the harmonic mean of the precision and recall scores. If we note TP the number of true positive predictions and do the same for FN standing for false negative predictions and FP for false positive predictions, the various scores introduced before are expressed as follow:

$$\begin{aligned} \text{Precision} &= \frac{TP}{TP+FP} \qquad & \text{F1-score} &= 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \\ \text{Recall} &= \frac{TP}{TP+FN} & &= \frac{2TP}{2TP+FP+FN} \end{aligned} \tag{12}$$

The recall defines the percentage of positive labels well predicted by the model, while the precision is the percentage of positive predictions accurately predicted. These two scores provide information about how the model handles the false positives (precision) and false negatives (recall), focusing then on the positive class. Since the F1-score represents the balance between recall and precision, it is the metric selected for analyzing our performance on the tumor detection tasks as the positive class is natural defined. However, for cancer classification, the positive class cannot be assigned to one specific cancer. This necessitates computing the F1-score for both the "0" class and the "1" class, as shown in Figure 11.

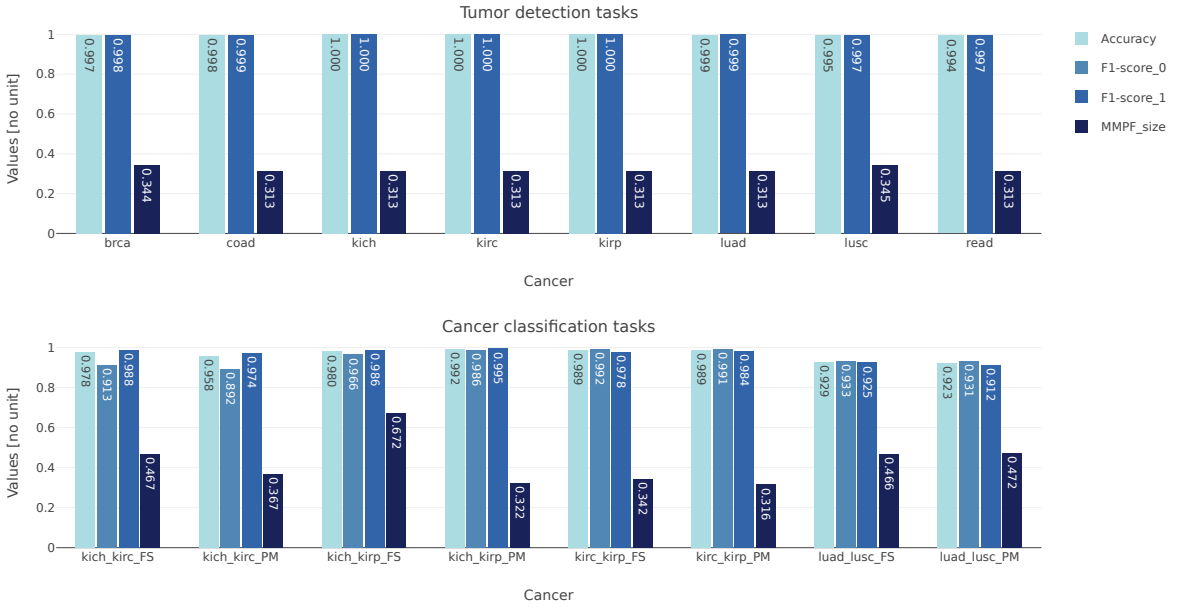


Figure 11 – Performances of the Baseline method on the tumor detection and cancer classification tasks evaluated on test sets with accuracy, F1-score on class 0 and 1, and the $\text{MMPF}_{\text{size}}$ metric.

As expected for tumor detection, the Baseline method gives almost perfectly accurate predictions on the test set for all the tasks. Based on this observation, the tumor detection tasks are considered fair enough with this model to be removed from the experiments with future methods. Indeed, the study of potential unfairness is hard to interpret when there is only one patient misclassified.

On the other hand, cancer classification tasks seems to be harder for the Baseline method, but without being less performing neither; the model remains very successful on these tasks too. Since the $\text{MMPF}_{\text{size}}$ metric is not interpretative directly from theory, the focus is only the accuracy and the F1-scores on 0 and 1 classes. As said before, the (un)fairness status of a method is highly subject to interpretation. It is why a deeper study of the results for this method is done later in Section 5 with another method as comparison.

4.2 Empirical Differential Fairness method

In this section, we present the work made in [20] and [21] about the definition of an ϵ-differentially fair mechanism and the addition of a penalty term to the loss function to rely this definition. They provide no specific architecture for an ϵ-differentially fair mechanism, then we simply use the same architecture presented for the Baseline method.

4.2.1 Presentation of the training method

Definiton A mechanism $h \in \mathcal{H}$ is said ϵ-differentially fair (ϵ-DF) with respect to A if for $y \in \text{range}(h)$, we have:

$$e^{-\epsilon} \leq \frac{\mathbb{P}_h(h(x) = y|a_i)}{\mathbb{P}_h(h(x) = y|a_j)} \leq e^{\epsilon}, \tag{13}$$

for all pairs of subgroups $(a_i, a_j) \in A \times A$.

The idea behind this notion of ϵ-DF is to assess for any subgroup that the probability for the outcomes to be right is similar to the others. To estimate the probability $\mathbb{P}_h$ for a classifier h, we simply use the empirical distribution of the outcomes from the model. We then note $\mathbb{P}_{\text{data}}(y|a) = \frac{N_{y,a}}{N_a}$, where $N_{y,a}$ is an empirical count of the samples from subgroup a associated with class y. And we can introduce the notion of Empirical Differential Fairness (EDF) corresponding to satisfy for all y, a_i and a_j the following inequality:

$$e^{-\epsilon} \leq \frac{N_{y,a_i}}{N_{a_i}} \frac{N_{a_j}}{N_{y,a_j}} \leq e^{\epsilon} \tag{14}$$

In the case where some subgroups has no patients for the class y (i.e. when $N_{y,a_j} = 0$), we need to define a stable version of the previous criterion (14), denoted smooth EDF. For that, the classifier probability prediction output $\mathbb{P}(h(x) = y|a)$ is used instead of empirical counts and we add a term α to avoid division by zero. For a positive class y and for all pairs of subgroups $(a_i, a_j) \in A \times A$, the smooth EDF is defined as follow:

$$e^{-\epsilon} \leq \frac{\sum_{x \in \mathcal{X}, A = a_i} \mathbb{P}(y|a) + \alpha}{N_{a_i} + |\mathcal{Y}|\alpha} \frac{N_{a_j} + |\mathcal{Y}|\alpha}{\sum_{x \in \mathcal{X}, A = a_j} \mathbb{P}(y|a) + \alpha} \leq e^{\epsilon} \tag{15}$$

Using Formula (15), we note $\epsilon_y(h)$ the minimum value of ϵ satisfying the inequality (15) for all subgroups a and for fixed model $h \in \mathcal{H}$ and class $y \in \{0, 1\}$. They propose then to construct the penalty term $R_{\text{pos}} = \max(0, \epsilon_1(h) - \theta)$, where the value of θ penalizes more or less ϵ-DF depending of its value, but set to 0 in the scope of the project. Finally, we obtain the loss function (16), which the training loop attempts to minimize.

$$\mathcal{L}_{\text{EDF}}(\tilde{y}, y) = \mathcal{L}(\tilde{y}, y) + \lambda R_{\text{pos}}, \tag{16}$$

where λ is an hyperparameter setting how much we consider the penalty term during the train and $\mathcal{L}$ is the classic Cross Entropy loss function. As a notice, we have that the penalty term is well differentiable using backpropagation method. And due to the setting $\theta = 0$, we can express $R_{\text{pos}} = \epsilon_1(h)$ (as ϵ is always positive), thus relieving concerns about vanishing gradients issue during the training, which could be caused by the initial max function.

The authors of this research paper propose this solution by focusing solely on a positive class $y = 1$. However, as mentioned earlier, this approach is not appropriate for cancer classification where the notion of positive or negative class is unclear. The following subsection will discuss this issue.

4.2.2 Different approaches for the penalty term's computation

To assess the restriction of the proposed penalty term to the positive class, we propose two different versions of this penalty term. The first one comes directly, instead of restricting to a specific class the metric, the ϵ value is the minimum value satisfying the inequality (15) for all class values $y = 0, 1$. More formally, we note $R_{\max}$ the penalty term restricted to the both positive and negative classes and it is expressed as follow:

$$R_{\max} = \max(\epsilon_0(h), \epsilon_1(h)) \tag{17}$$

After that we replace this new penalty term in the equation (16) to obtain the expression of the new loss function. The second idea, is to consider the two classes at the same time. Instead of choosing the worst one between the negative or the positive class, we keep track of the both. We come then with the penalty term R_{sum} expressed as follow:

$$R_{\text{sum}} = \frac{1}{2}(\epsilon_0(h) + \epsilon_1(h)) \tag{18}$$

This idea behind this last proposition may be difficult to apprehend intuitively, but we aimed to consider both classes so that updates to the weights are made with respect to both class 0 and 1 at each iteration of training. Consequently for the EDF method, we have three different ways for computing the EDF penalty term. We will compare the efficiency of these penalty term computations in the next section.

4.2.3 Selection of the hyperparameters of the method regarding its performances

The training method for the EDF method is similar to that used in the Baseline method, with the only difference lying in the loss function, where we add the discussed penalty term using various ways of computation. As mentioned earlier, we now only focus on the cancer classification tasks, for which all three penalty term computation methods are evaluated for training process. Table 2 summarizes all the evaluations with the $\text{MMPF}_{\text{size}}$ metric on validation sets of the different tasks trained with different penalty terms.

Table 2 – Cancer classification tasks evaluation with $\text{MMPF}_{\text{size}}$ metric on validation set with all three penalty computation methods: R_{pos}, $R_{\max}$ and R_{sum}.

Penalty term computation methods		R_{pos}	$R_{\max}$	R_{sum}
Cancer classification	kich_kirc_FS	**0.47**	0.58	**0.47**
	kich_kirc_PM	**0.39**	**0.39**	**0.39**
	kich_kirp_FS	**0.63**	**0.63**	**0.63**
	kich_kirp_PM	**0.33**	**0.33**	**0.33**
	kirc_kirp_FS	**0.33**	**0.33**	**0.33**
	kirc_kirp_PM	**0.43**	**0.43**	**0.43**
	luad_lusc_FS	**0.52**	**0.52**	**0.52**
	luad_lusc_PM	**0.49**	**0.48**	**0.49**

Observations from Table 2 indicates that the computation method for the penalty term does not change that much the results for most of the classification tasks. The only difference appears for the cancer classification task on kich_kirc_FS, where the $R_{\max}$ metric gives worst results in terms of fairness. Then, all the tasks are now being evaluated along the accuracy, the F1-scores on classes 0 and 1, and the $\text{MMPF}_{\text{size}}$ metric on the test sets using the most adapted computation method for the penalty term represented by R_{pos} in our case. The results are summarized on Figure 12.

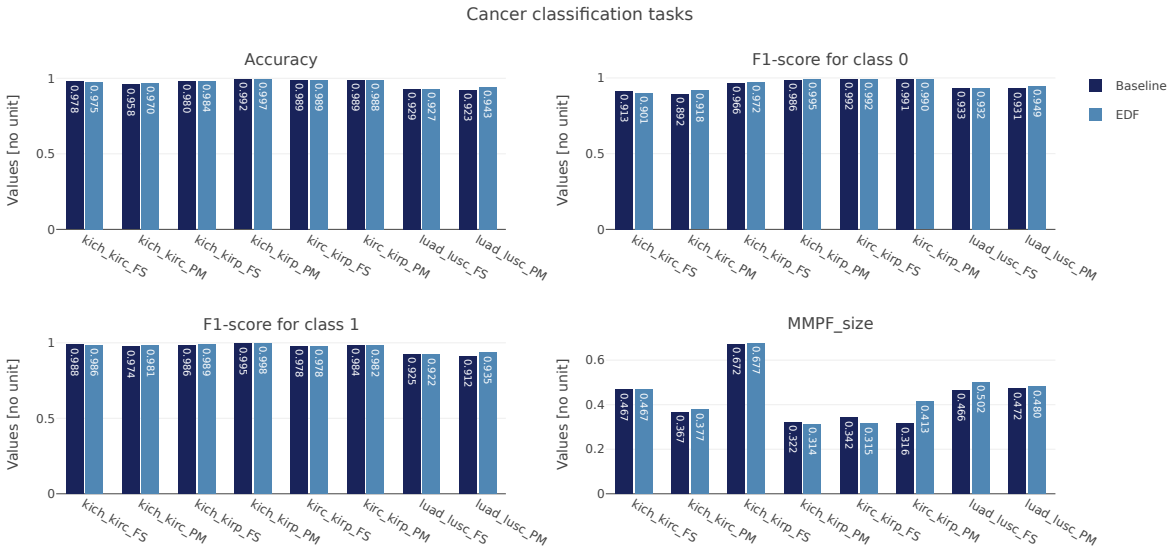


Figure 12 – Performances of the EDF method compared to those of the Baseline method measured with the accuracy, F1-score for class 0 and 1, and the $\text{MMPF}_{\text{size}}$ metric evaluated on test sets.

Upon observing the results depicted in Figure 12 for the accuracy and the F1-scores, we can see a similarity in performance compared to those obtained with the Baseline method. When looking at the $\text{MMPF}_{\text{size}}$ metric, the results are more nuanced. While there is no difference for the tasks kich_kirc_FS and kich_kirp_FS, the EDF method seems to have better performances (reminder: lower the better for this metric) on the tasks kich_kirp_PM and kirc_kirp_FS. But it gives worst results for the remaining tasks. However, our intention here is not to delve deeper into the results of each method. Rather, we simply make sure the model is performing correctly. A more in-depth analysis of the results is done in Section 5.

4.3 Two players game formulation of the intersectional fairness problem - MinimaxFair

In the following section, we present the work made in [22] about a two-player game formulation of the intersectional fairness problem. The idea is to use a re-formulation of the intersectional fairness problem by introducing the MMPF metric and next use an re-weighting method on the different subgroups of the data set during the training. And as before, the architecture of the model doesn't change and stays as the one use for the Baseline method.

4.3.1 Presentation of the training method

Before introducing the re-weighting method given here, we recall the formulation of the intersectional fairness problem as presented in [22] and introduced in Section 3.2.

$$h^* = \arg\min_{h \in \mathcal{H}} \max_{a \in A} r_a(h) \tag{19}$$

The method proposed here is a zero-sum game method between a learner and a regulator, named MinimaxFair. The idea is to use the Baseline method as an oracle process (solver) that finds an optimal model $h_n \in \mathcal{H}$ for a fix weights vector $\omega \in [0,1]^{N_A}$ at each iteration $n \in \mathbb{N}$. We note this oracle process as Baseline(ω) that returns a model h_n. But the Baseline method is modified on the loss function where the loss is weighted along the different subgroups with ω. It gives the following formulation:

$$\begin{cases} h_n = \text{Baseline}(\omega) := \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{N_A} \omega_i \mathbf{r}_{a_i}(h) \\ \|\omega\|_1^1 = 1, \ \ \omega_i > 0 \ \ \forall i \in \{1, \dots, N_A\} \end{cases} \tag{20}$$

Then at each iteration n of the process, the regulator computes a weighting over the different subgroups using the well-known Exponential-Weights algorithm [23] with respect to the errors achieved by the model h_n. The coefficient η_n for the Exponential-Weights algorithm is updated at each iteration n to $\frac{1}{\sqrt{n}}$. With this game method, the empirical average of the plays achieves a $\frac{1}{\sqrt{n}}$-approximate Nash equilibrium [24].

This method gives us a sequence $h_1, \dots, h_{N_{\text{iter}}}$ where the one with the smallest error group rate on the validation set is kept as the final model. More formally, the method can be expressed with Algorithm 2.

Algorithm 2: MinimaxFair

Input Number of iterations N_{iter} ;
$r_a(h) = \frac{1}{|a|} \sum_{(x,y) \in a} \mathcal{L}(h(x), y)$;
Initialization of ω with $\omega_k = \frac{1}{a_k}$ for $k = 1, \dots, N_A$;
for $n = 1, \dots, N_{iter}$ **do**
  Find $h_n = \text{Baseline}(\omega) := \arg\min_{h \in \mathcal{H}} \sum_{k=1}^{N_A} \omega_k r_{a_k}(h)$;
  Update the term $\eta_n = \frac{1}{\sqrt{n}}$;
  Update the subgroups weights vector ω such that $\omega_k = \omega_k \exp(\eta_n r_{a_k}(h_n))$;
  if $\max_{a \in A} r_a(h_n) < r^*$ **then**
    Save h_n as h^* and $\max_{a \in A} r_a(h_n)$ as r^*;
  end
end
Output Best model h^*

4.3.2 Performances of the model on the different datasets

Using the same setup as for Baseline and EDF methods, the MinimaxFair method is evaluated on the various cancer classification tasks with accuracy, F1-scores for class 0 and 1, and $\text{MMPF}_{\text{size}}$ metric. The results are depicted in Figure 13.

In terms of overall performance defined by the accuracy and the F1-scores, as depicted in Figure 13, the MinimaxFair method shows similar performance to that of the Baseline method, except in the case of luad_lusc_FS, where it appears slightly inferior. Regarding the results of the $\text{MMPF}_{\text{size}}$ metric, the MinimaxFair method seems to manifest either less fairness or comparable fairness to the Baseline across most of the cancer classification tasks. The only improvement can be observed for the kich_kirp_PM task. However, as mentioned earlier, understanding the significance of these metric values in terms of actual fairness improvement requires further analysis, which we present in Section 5.

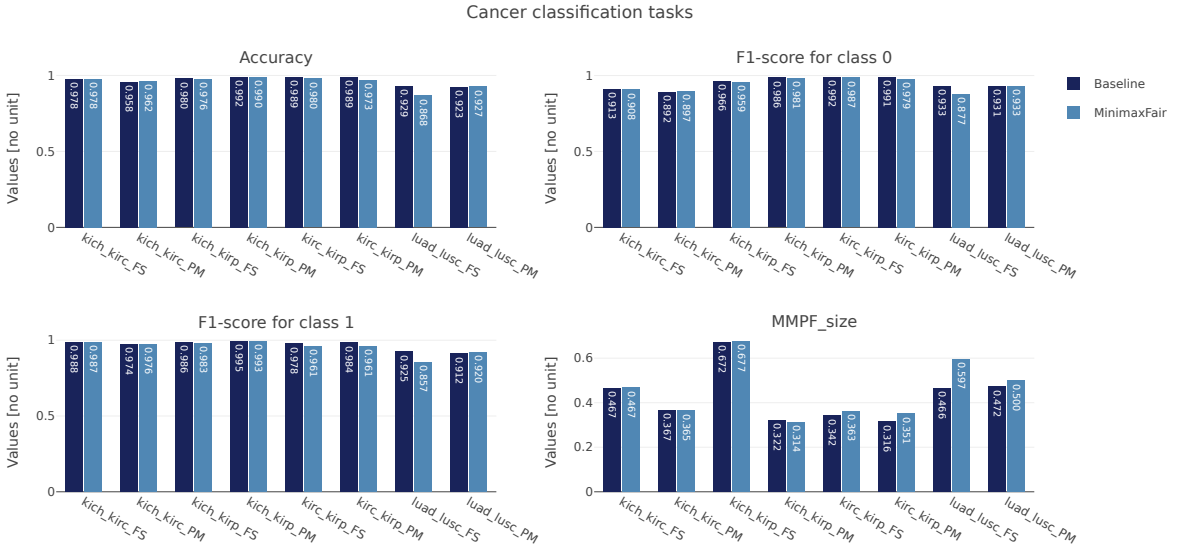


Figure 13 – Performances of the MinimaxFair method compared to those of the Baseline method measured with the accuracy, F1-score for class 0 and 1, and the $\text{MMPF}_{\text{size}}$ metric evaluated on test sets.

4.4 Approximate Projection onto Star sets - APStar

Through this section, we present the work made in [8] where they present a new algorithm to assess intersectional fairness problem: Approximate Projection onto Star set (APStar). This algorithm follows the same idea as presented for MinimaxFair by re-weighting the loss computed for each subgroup. The difference appears for the method used to update the weights at each iteration.

4.4.1 Presentation of the training method

To explain how they come up with this method, we must introduce some notions before.

Definition A vector $\mathbf{r}' \in \mathbb{R}^{N_A}$ is said to dominate another vector $\mathbf{r} \in \mathbb{R}^{N_A}$, noted $\mathbf{r}' \prec \mathbf{r}$ if $r'_i \leq r_i$ for all $i = 1, \ldots, N_A$ with a strict inequality for at least one $j \in \{1, \ldots, N_A\}$. And we note $\mathbf{r}' \preceq \mathbf{r}$ if $\mathbf{r} \not\prec \mathbf{r}'$.

Definition A classifier or model $h' \in \mathcal{H}$ is said to dominate h, noted as $h' \prec h$, if the group-specific risks satisfies $\mathbf{r}(h') \prec \mathbf{r}(h)$. Similarly, we note $h' \preceq h$ if $\mathbf{r}(h') \preceq \mathbf{r}(h)$.

Definition Given the group-specific risk functions $\mathbf{r}(h)$ and the family of classifiers $\mathcal{H}$, the set of Pareto front classifiers is expressed as $\mathcal{P}_{A,\mathcal{H}} = \{h \in \mathcal{H} | h \preceq h', \forall h' \in \mathcal{H}\}$. The corresponding achievable risks are denoted as $\mathcal{P}^{\mathcal{R}}_{A,\mathcal{H}} = \{\mathbf{r} \in \mathbb{R}^{N_A} | \exists h \in \mathcal{P}_{A,\mathcal{H}}, \mathbf{r} = \mathbf{r}(h)\}$.

As before, we consider a weights vector $\omega \in [0,1]^{N_A}$ and we want to solve the following linear weighting problem:

$$\begin{cases} \hat{h} = \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{N_A} \omega_i r_{a_i}(h) \\ \|\omega\|_1^1 = 1, \ \ \omega_i > 0 \ \ \forall i \in \{1, \ldots, N_A\} \end{cases} \tag{21}$$

We note h_ω the solution of Problem (21) for a fixed weights vector ω. And to lighten the notations, we write $\mathbf{r}_\omega$ as the group-specific risk $\mathbf{r}(h_\omega)$. Using the results derived in [25], they prove the following theorem.

Theorem Given a convex classifier's class $\mathcal{H}$ and $\mathbf{r}(h)$ convex risk function, then:

1. The Pareto front is convex regarding the dominance notion introduced before, i.e. : $\forall \mathbf{r}, \mathbf{r}' \in \mathcal{P}^{\mathcal{R}}_{A,\mathcal{H}}, \ \ \lambda \in [0,1], \ \ \exists \mathbf{r}'' \in \mathcal{P}^{\mathcal{R}}_{A,\mathcal{H}} : \mathbf{r}'' \preceq \lambda \mathbf{r} + (1-\lambda)\mathbf{r}'$.
2. Every Pareto solution is a solution of the problem (21), i.e. : $\forall \hat{\mathbf{r}} \in \mathcal{P}^{\mathcal{R}}_{A,\mathcal{H}}, \ \ \exists \omega : \hat{\mathbf{r}} = \mathbf{r}_\omega$.

All these notions are meant to get a better understanding of the theorem that comes next. It shows the important properties that must satisfy a vector ω' to improve the minimax risk obtained by a previous vector ω at any given iteration (i.e. $\|\mathbf{r}_{\omega'}\|_\infty < \|\mathbf{r}_\omega\|_\infty$). It mainly defines the re-weighting method presented for this method.

Theorem Under the following conditions:

1. $\omega' \notin \arg\min_{\omega \in \Delta^{N_A}} \|\mathbf{r}(\omega)\|_\infty$
2. $\omega^* \in \arg\min_{\omega \in \Delta^{N_A}} \|\mathbf{r}(\omega)\|_\infty$
3. $N_i = \{\omega : r_i(\omega) < \|\mathbf{r}(\omega')\|_\infty\}$ for $i = 1, \ldots, N_A$

We have the following statements:

1. $\omega^* \in \bigcap N_i$
2. For $i = 1, \ldots, N_A$, if $\omega \in N_i$ then for all $\lambda \in [0,1]$, we have the following: $\lambda\omega + (1-\lambda)e_i \in N_i$, where e_i is the i-th element from the standard basis in $\mathbb{R}^{N_A}$
3. For any sub-partition $\mathcal{I} \subseteq \{1, \ldots, N_A\}$ and any weight vector ω such that $\omega_i = 0 \ \ \forall i \in \mathcal{I}$, then we have that $\omega \in N_i$
4. If the risk function $\mathbf{r}(\omega)$ is also continuous in ω, then for all sub-partition $\mathcal{I} \subseteq \{1, \ldots, N_A\}$ such that $\omega \in \bigcap_{i \in \mathcal{I}} N_i$, we have that there exist $\epsilon > 0 : \ B_\epsilon(\omega) \subset \bigcap_{i \in \mathcal{I}} N_i$
5. If the achievable risks set $\mathcal{P}^{\mathcal{R}}_{A,\mathcal{H}}$ is convex, the solution $\mathbf{r}_{\omega^*} \in \arg\min_{\mathbf{r} \in \mathcal{P}^{\mathcal{R}}_{A,\mathcal{H}}} \|\mathbf{r}\|_\infty$

Using these statements, we construct the Approximate Projection on Star sets (APStar) Algorithm 3 with a specific re-weighting method for ω. The intuition behind this process, for a fixed linear weights vector ω_n at iteration n, is to identify the subgroups indices where it reduces the risk as a subset $\mathcal{I} \subset [N_A] := \{1, \ldots, N_A\}$. More formally, we identify the subset $\mathcal{I}$ where $\omega_n \in \bigcap_{i \in \mathcal{I}}$. Doing so, we use ω_n and $\omega_{[N_A] \setminus \mathcal{I}} \in \bigcup_{i \in [N_A] \setminus \mathcal{I}} N_i$ (unsatisfied group risks) and by getting a linear interpolation between these vectors to generate a new vector $\omega_{n+1} = \alpha\omega_n + (1-\alpha)\omega_{[N_A] \setminus \mathcal{I}}$. The coefficient α is set to 0.5 as default value and is kept that way. Algorithm 3 is an overview of the pseudo code used to train the model.

Algorithm 3: APStar

Input Number of iterations N_{iter} ;
$r_a(h) = \frac{1}{|a|}\sum_{(x,y)\in a}\mathcal{L}(h(x), y)$;
Initialization
ω with $\omega_k = \frac{1}{a_k}$ for $k = 1, \ldots, N_A$;
$h_0, \mathbf{r}_\omega \leftarrow \text{Baseline}(\omega) := \arg\min_{h\in\mathcal{H}} \sum_{k=1}^{N_A} \omega_k r_{a_k}(h)$;
$\bar{r} \leftarrow \|\mathbf{r}_\omega\|_\infty$;
for $n = 1, \ldots, N_{iter}$ **do**
  $\mathbf{1}_\omega \leftarrow \{\mathbf{1}(r_i(\omega) \geq \bar{r})\}_{i=1}^{N_A}$;
  $\omega \leftarrow (\alpha\omega + \frac{1-\alpha}{n\|\mathbf{1}_\omega\|_1^1}\mathbf{1}_\omega)\frac{n}{(n-1)\alpha+1}$;
  $h_n, \mathbf{r}_\omega \leftarrow \text{Baseline}(\omega) := \arg\min_{h\in\mathcal{H}} \sum_{k=1}^{N_A} \omega_k r_{a_k}(h)$;
  if $\|\mathbf{r}_\omega\|_\infty < \bar{r}$ **then**
    $\bar{r} \leftarrow \|\mathbf{r}_\omega\|_\infty$;
    $h^*, \mathbf{r}^* \leftarrow h_n, \mathbf{r}_\omega$;
  end
end
Output $h^*, \mathbf{r}^*$

4.4.2 Performances of the model on the different datasets

Using the same setup as for Baseline, EDF and MinimaxFair methods, the APStar method is evaluated on the various cancer classification tasks with accuracy, F1-scores for class 0 and 1, and $\text{MMPF}_{\text{size}}$ metrics. The results are depicted in Figure 14.

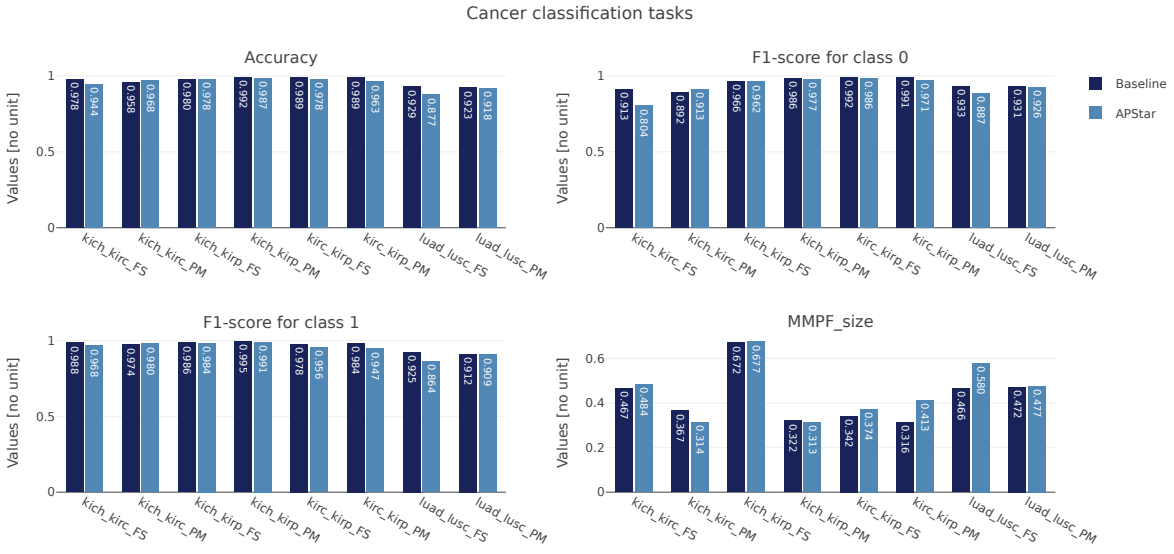


Figure 14 – Performances of the APStar method compared to those of the Baseline method measured with the accuracy, F1-score for class 0 and 1, and the $\text{MMPF}_{\text{size}}$ metric evaluated on test sets.

The performances of the APStar method compared to the Baseline method appears to be inferior for kich_kirc_FS, kirc_kirp_PM, luad_lusc_FS and luad_lusc_PM tasks, or at most equivalent for the remaining tasks, as measured by accuracy and F1-scores. However, the results become more nuanced when considering the $\text{MMPF}_{\text{size}}$ metric. While the APStar method indicates poorer fairness performance for most of the tasks, it seems to give improvements for the kich_kirc_PM and kich_kirp_PM tasks. To have a better understanding of the significance of these metric values in terms of intersectional fairness improvement, we delve into the results of all the proposed methods tested for the different cancer classification tasks in following Section 5.

5 Comparison of the performances of the different methods

The methods being introduced and trained at their best performances using hyperparameters tuning on validation sets, we compare these performances EDF, MinimaxFair and APStar methods to Baseline method to identify potential improvements on the intersectional fairness aspect. The results obtained on the test sets regarding the $\text{MMPF}_{\text{size}}$ metric are summarized in Table 3, where the best value obtained for each cancer classification task is highlighted. As a reminder, this last metric indicates a fairer model when evaluating a lower value on it.

Table 3 – Comparison of the performances of the four proposed methods (Baseline, EDF, MinimaxFair and APStar) regarding the $\text{MMPF}_{\text{size}}$ metric across the various cancer classification tasks.

Training method		Baseline	EDF	MinimaxFair	APStar
Cancer classification	kich_kirc_FS	**0.47**	**0.47**	**0.47**	**0.48**
	kich_kirc_PM	0.37	0.37	0.37	**0.31**
	kich_kirp_FS	**0.67**	**0.68**	**0.68**	**0.68**
	kich_kirp_PM	**0.32**	**0.31**	**0.31**	**0.31**
	kirc_kirp_FS	0.34	**0.31**	0.36	0.37
	kirc_kirp_PM	**0.32**	0.41	0.35	0.41
	luad_lusc_FS	**0.47**	0.50	0.60	0.58
	luad_lusc_PM	**0.47**	0.48	0.50	**0.48**

A direct observation depicted in Table 3 is the out-performances of the Baseline method regarding the $\text{MMPF}_{\text{size}}$ metric as an intersectional fairness indicator. Indeed, regarding this metric, the Baseline method outperforms or at least matches the performance of the other three methods in 6 of the 8 cancer classification tasks. Additionally, the MinimaxFair method achieves its best performances in 3 of these 6 tasks that are best performed by the Baseline. Therefore, we assess the use of the MinimaxFair method is at most equivalent of using the Baseline one applied to our cancer classififcation tasks.

For the APStar method, the results are more nuanced. Indeed, it gives performances equivalent to the ones given by Baseline for 4 tasks: kich_kirc_FS, kich_kirp_FS, kich_kirp_PM and luad_lusc_PM. The only task where the APStar method outperforms the Baseline is for kich_kirc_PM. Since the behaviour of the metric is unknown, i.e. we do not know how significant is this improvement, we can not assess if the use of this method is adapted specifically to this task.

And for the EDF method, the interpretation of the results is the similar to APStar. Indeed, this method achieves similar performances as the Baseline method for kich_kirc_FS, kich_kirp_FS and kich_kirp_PM, and outperforms it for kirc_kirp_FS. But as said before, we can not assess whether the use of this method is adapted specifically to this task. To do so, we give an in-depth analysis of the

results observed for kirc_kirp_FS with the Baseline and EDF methods.

For this analysis, the test set from kirc_kirp_FS dataset is studied at the subgroup level. As a reminder, the results depicted in Figure 12 for the $\text{MMPF}_{\text{size}}$ metric indicates an improvement from 0.342 for the Baseline method to 0.315 for the EDF method. On the left heatmaps in Figure 15 is illustrated the distribution of patients among subgroups, where on the upper side we have all the females and the males on the other side. The distribution is important here to interpret the behaviour of the $\text{MMPF}_{\text{size}}$ metric that depends of the number of patients there is in each subgroup. For the case of the test set of kirc_kirp_FS, the $\text{MMPF}_{\text{size}}$ metric focus on the average loss of all the subgroups with more than 11 patients in it. This average loss of each subgroup given the predictions of the Baseline method is depicted on the two centered heatmaps in Figure 15. And for the EDF method, there are depicted in the heatmaps on the right of Figure 15.

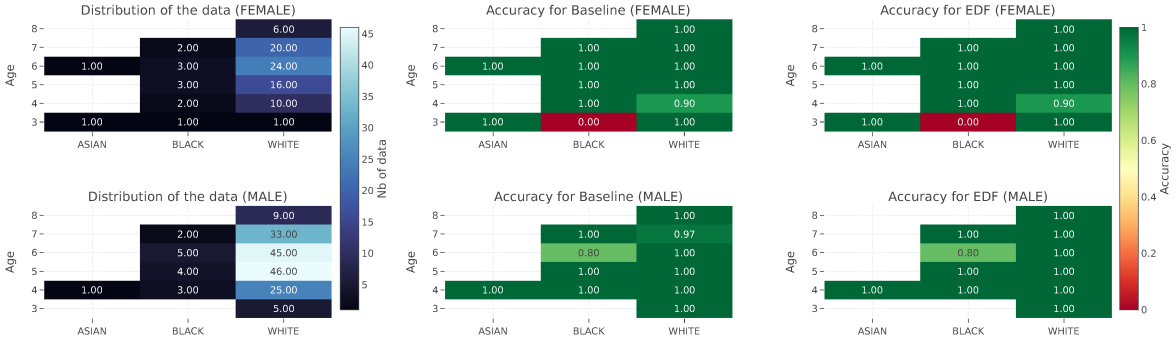


Figure 15 – In-depth analysis of the predictions given by Baseline and EDF methods on test set of the classification task kirc_kirp_FS.

The first observation from this figure is the sparsity of the distribution of the data among the subgroups similar to the one observed in Figure 3. It implies that many subgroups (all the ones with less than 11 patients) are going to be removed from the scope of the $\text{MMPF}_{\text{size}}$ metric.

The second observation is the very good performances of the Baseline method. Indeed, for 268 patients given in this test set, only 4 are misclassified by the model. This leads us to think that being unfair on such a well-performing model is hard to define clearly and maybe we already reached the maximum performances possible for this task.

And finally, when looking at the results proposed by the EDF method, we observe an improvement by being accurate on the patient wrongly classified by Baseline belonging to the subgroup characterized by three protected protected attributes fixed to: 70-79 for the age, white for the race and male for the gender. Then, we can consider the EDF model as fairer than Baseline by being more accurate, but Baseline can not be considered as unfair neither. We can not make an assessment of intersectional fairness on the predictions coming from one only patient.

Moreover, suppose the situation where the metric is considering all the subgroups with more than 5 patients. The metric would then consider the subgroup with accuracy 0.80 for the both methods (60-69,

black and male) and then consider it as the worst performing subgroups. And instead of giving similar metric values for the two methods as expected, the metric gives 0.55 for the Baseline and 0.49 for the EDF. Which indicates a better improvement than the one observed earlier, which is not. The issue here comes from the use of the Cross-Entropy Loss function (11) inside the computation process of the $\text{MMPF}_{\text{size}}$ metric. Indeed, this function is highly sensitive to the uncertainty of the model predictions, since it takes as input the raw probabilities of the sample of belonging to one or the other class.

6 Remove the protected attributes from input to the model

6.1 Introduction

An assessment we made thus far concerns the availability of the different protected attributes from each patient. In many situations, such information is hidden and then not accessible. And with the actual models we have proposed, the protected attributes of each sample are required as input for classification. The challenge is then to remove these protected attributes (age, race and gender) from the input provided to the models. Apart the technical modifications this implies, there is also the challenge of removing information that could serve as a key indicator of the patient's subgroup affiliation. The model would then have to identify subgroup-related information among the 768 features obtained from the histopathological whole-slide images to asses potential intersectional fairness issue in the predictions.

Regarding the technical modifications, the shape of each data sample is reduced from $N_{\text{slices}} \times 771$ to $N_{\text{slices}} \times 768$ by removing the expanding step depicted in Figure 4. These changes do not directly impact the method to split the dataset into training, validation and test subsets. But we continue to track the demographic information of each sample to maintain consistent distribution ratios among the subgroups across the train, validation and test sets. However, this information is not added to the final data tensors, as explained earlier.

For the training methods themselves, there are no particular changes except for adjusting the size of the linear layers of the model for the new shape of the data. The main consideration with this modification lies in the EDF, MinimaxFair and APStar methods, which still require access to subgroup information for the loss computation necessary for their methods, unlike the Baseline. These three methods need access to the patient's subgroup affiliation to accurately update the various weights they utilize. However, this information is only used during model training and still ensures data inference without any protected attributes information. Since the Baseline method has no access to demographic information about the patient, it may be expected to underperform compared to the other three methods. Furthermore, the selection of hyperparameters and the validation steps for the different methods remain unchanged as they are not affected by these changes.

Finally, we keep the $\text{MMPF}_{\text{size}}$ metric as the primary measure to compare the performances of the different training methods in terms of intersectional fairness. Consequently, we still need to track the protected attributes of each sample to know its belonging to various subgroups. But, this does not impact the evaluation of patients by the models without the demographic information as input.

6.2 Performances

The technical aspects of the modifications being presented, the different methods can be evaluated on the various cancer classification tasks of histopathological whole-slide images. All steps, including data splitting, training, hyperparameter selection and validation, of each method have been conducted the

same way (or with the modifications explained earlier) to obtain the following results. These results are evaluated on test sets for each classification task. In Figure 16, the accuracy and the $\text{MMPF}_{\text{size}}$ metric are depicted for each of the four training methods: Baseline, EDF, MinimaxFair and APStar.

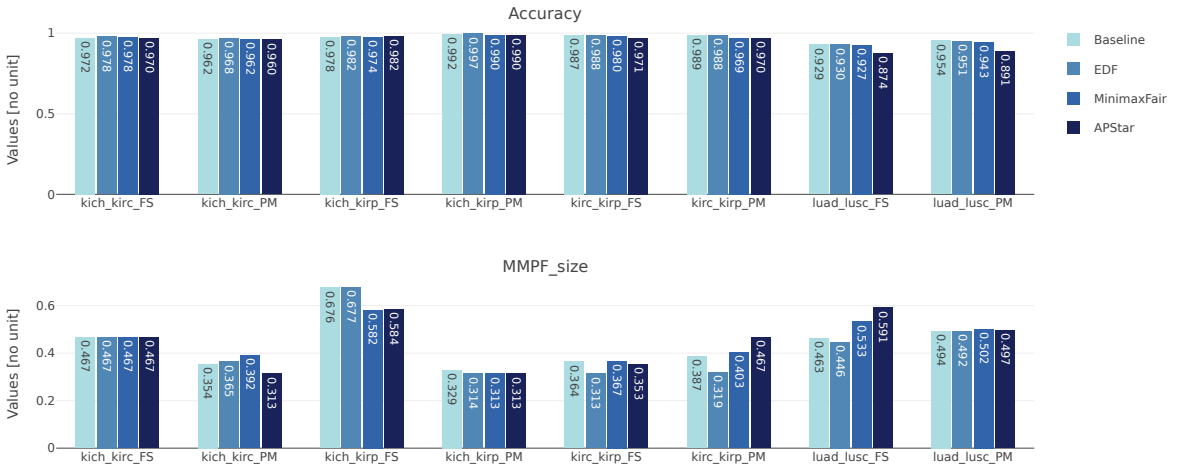


Figure 16 – Accuracy and $\text{MMPF}_{\text{size}}$ metric performances of the Baseline, the EDF, the MinimaxFair and the APStar methods evaluated on the 8 different cancer classification tasks.

Upon observing the results depicted in Figure 16, the Baseline method again shows very good performances on the multiples cancer classification tasks. The worst-performing task for this method is for luad_lusc_FS, where it reaches a 92.9% regarding the accuracy, or for kich_kirc_PM, where it reaches a score of 0.676 regarding the $\text{MMPF}_{\text{size}}$ metric.

For the results concerning the EDF method, it gives similar results to the Baseline method in term of accuracy across all the 8 cancer classification tasks. However, this differs for the $\text{MMPF}_{\text{size}}$ metric, where the MinimaxFair method seems to be as fair as the Baseline for half of the tasks and appears to be fairer for kich_kirp_PM, kirc_kirp_FS, kirc_kirp_PM and luad_lusc_FS tasks.

The MinimaxFair method, like the EDF method, shows similar results to the Baseline method in term of accuracy across all the 8 cancer classification tasks. However, there is a notable difference for the $\text{MMPF}_{\text{size}}$ metric, where the MinimaxFair method seems to give a significant improvement for the kich_kirp_FS task. Specifically, there is a decrease from 0.676 for the Baseline method to 0.582 for the MinimaxFair method.

Finally, for the APStar method, the accuracy indicated for most tasks follows that of the Baseline method, except for the luad_lusc tasks (FS and PM), where the APStar method is clearly underperforming. Nevertheless, the APStar method shows a similar range of improvements as the MinimaxFair method for the kich_kirp_FS task compared to the Baseline method. This potential improvement in terms of intersectional fairness will be analyzed in depth.

With the results of all the different training methods presented, we now delve into the results provided by the BASELINE and APSTAR methods for the cancer classification task kich_kirp_FS. The idea here is to conduct a similar study as in the previous Section 5. We depict the distribution of patients among the different subgroups characterized by the age, the race and the gender (upper side for female and lower side for male) on the left side of Figure 17. The accuracy of predictions given by the BASELINE and APSTAR methods at the subgroup level is depicted respectively in the center and right sides of Figure 17.

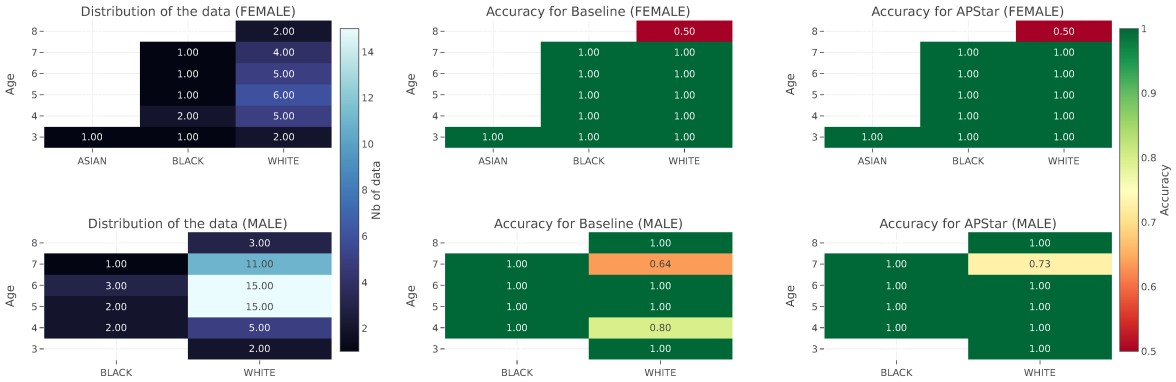


Figure 17 – In-depth analysis of the predictions given by BASELINE and APSTAR methods on test set of the classification task kich_kirp_FS.

The first observation from this figure is the unequal distribution of data among the multiple subgroups. Indeed, the $\text{MMPF}_{\text{size}}$ metric is designed to ignore subgroups with strictly less than 4 samples. Thus, the worst-performing subgroup (80-89 for age, white for race and female for gender) is ignored in the metric computation.

A second observation is only about the BASELINE method, which gives relatively good performances for this task (97.8% of accuracy). However, it still struggles to give accurate predictions on the specific subgroup characterized by 70-79 for the age, white for the race and male for the gender. Indeed, the model reaches an accuracy of 64% on this subgroup, which comprises 11 patients. Yet, given the small sample size of 11 patients, giving conclusions about unfairness for this subgroup is challenging.

Finally, comparing with the APSTAR method, notable improvements are observed for the previously mentioned subgroup, where we jump from an accuracy of 64% for the BASELINE to an accuracy of 73% for the APSTAR. Furthermore, the last subgroup, characterized by 40-49 for the age, white for the race and male for gender, is now accurately predicted with the new method. Unlike the previous situation from Section 5, there are more indications of improvement in terms of intersectional fairness. Moreover, this improvement is achieved without causing unnecessary harms to the best-performing subgroups. However, it's essential to note that these improvements are based on predictions made accurately for only 5 patients, which may not be sufficient to conclude that the APSTAR method outperforms the BASELINE, especially considering its performance across the other cancer classification tasks.

7 Discussion

Through this report, we explored various training methods to assess potential intersectional fairness issues employing a modified version of the MiniMax Pareto Fairness (MMPF) metric proposed in different research papers.

Firstly, we presented 4 variations of the original MMPF metric: MMPF_5, MMPF_{10}, $\text{MMPF}_{\text{adapted}}$ and $\text{MMPF}_{\text{size}}$. These vaiations aimed to address the issue of very small subgroups (e.g. with only one patient) present in the different histopathological images classification tasks. To select among these 5 metrics (including the original one), we proposed a random generator of fair and unfair scenarios similar to those expected to be encountered with these datasets. It then indicated a performing metric we choose to retain as our main comparison metric for future methods: the $\text{MMPF}_{\text{size}}$ metric. However, some limitations were later observed with this metric. One concerns the construction of the metric itself, which use the average Cross-Entropy Loss function (11) on each subgroup. As explained in the previous Section 5, this loss function is highly sensitive to the model's uncertainty regarding the predictions it makes. This sensitivity could lead to contradictory situations where the metric indicates a clear improvement of one method (A) over another method (B), when method A is actually equivalent to or even worse than method B in some cases. One solution to address this issue could be the use of a classification metric instead of a loss function. For example, the F1-score could replace this loss function and would not be sensitive to the uncertainty of the models.

Another limitation of this metric arises from the decision to exclude the small subgroups from the scope of the metric. This makes the metric highly specific to situations where the data are very unequally distributed among the subgroups. Essentially, the models are biased by the size of each of its subgroups during the training, potentially leading to situation where these models under-perform on small subgroups (i.e. being actually unfair) that the metric fails to capture. To address this, we introduced the concept of averaging performances across multiple subgroups as proposed for the MMPF_{10} or $\text{MMPF}_{\text{adapted}}$ metrics. However, a more in-depth analysis of these metrics is necessary to fully understand the boundary cases we may encounter.

On a second point, we introduced three different training methods to address potential issues of intersectional fairness. These methods all use the same model architecture and have been compared to a Baseline method to represent scenarios where unfair situations may arise. These methods are adapted variations of methods proposed in research papers [20], [22] and [8]. All four methods (including the Baseline) gave remarkably good performances across all different classification tasks, indicating their effectiveness. However, the three proposed ones did not surpass the Baseline, and when they did, the improvement was marginal, typically affecting only one patient. This observation suggests that the Baseline training method may have already achieved optimal performance on these tasks, without changing in-depth the architecture of the model.

One hypothesis to explain this issue is that the models had access to demographic information in the data, which could encode significant information about predictions and for which its access may not be guaranteed for future new data. Then, these protected attributes were removed from the data tensors provided to the models. The models were then trained in the same manner as before to assess the potential improvements these fairness methods could bring. The results are more nuanced compared to those with demographic information. However, they are not conclusive about potential unfairness issue for the Baseline method and improvements on this issue with the proposed EDF, MinimaxFair and APStar methods. The Baseline method remains the most consistent across the different cancer classification tasks.

Then, another plausible explanation could be that the issue lies not in the training process itself but in the data processing stage, particularly during feature extraction. The features are extracted from each slice of histopathilogical whole-slide images using the pre-trained CHIEF model, which is self-supervised trained on histopathological samples containing various cancers relevant to our classification task. Consequently, an hypothesis would be that this model has already learned features defining different cancers, or potentially even trained on data similar to what we have access to. It would then explain why the Baseline method performs such well on the cancer classification tasks. One potential solution could be using whole-slide images as input instead of pre-extracted features, but this would require significant changes of the model architecture and certain training methods.

Another approach we could have explored in the situation where there was a clear intersectional unfairness among the predictions given by the Baseline method, is to adapt the MinimaxFair and APStar methods using the proposed $\text{MMPF}_{\text{size}}$ metric. Indeed, the basiss of these training methods is the original MMPF metric, which we could replace with our new version to make it more relevant to the cancer classification tasks we are working with.

8 Conclusion

In conclusion, we explored various training methods and fairness metrics to ensure potential intersectional fairness in different classification tasks using histopathological whole-slide images. The introduction of modified versions of the MMPF metric aimed to address issues with small subgroups in the data distribution, but yet revealed sensitivity limitations. The comparison of different training methods against a baseline approach highlighted the challenge of improving upon already effective models. Despite efforts to remove demographic information, no conclusive evidence can be made regarding the unfairness of the Baseline method or the effectiveness of the proposed methods. This suggests that the issue lies not in the training process itself but in the data processing stage, particularly during feature extraction with the pre-trained CHIEF model. Addressing these challenges will require further investigation for the fairness metrics and potentially significant changes to model architecture and training methods to work with histopathological whole-slide images directly.

References

[1] Wang et al. "A Generalizable Foundation Model for Quantitative Pathology Image Analyses". In: (2024).

[2] Xiyue Wang, Sen Yang, Jun Zhang, Minghui Wang, Jing Zhang, Wei Yang, Junzhou Huang, and Xiao Han. "Transformer-based unsupervised contrastive learning for histopathological image classification". In: *Medical Image Analysis* 81 (2022), p. 102559. ISSN: 1361-8415. DOI: https://doi.org/10.1016/j.media.2022.102559.

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].

[4] *The Cancer Genome Atlas Program (TCGA)*. URL: https://www.cancer.gov/ccg/research/genome-sequencing/tcga.

[5] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. "Machine Bias: There's software used across the country to predict future criminals. And it's biased against blacks." In: *ProPublica* (2016). URL: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing/.

[6] Alex Beutel, Jilin Chen, Zhe Zhao, and Ed H. Chi. *Data Decisions and Theoretical Implications when Adversarially Learning Fair Representations*. 2017. arXiv: 1707.00075 [cs.LG].

[7] Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. "Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper_files/paper/2016/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf.

[8] Natalia Martinez, Martin Bertran, and Guillermo Sapiro. *Minimax Pareto Fairness: A Multi Objective Perspective*. 2020. arXiv: 2011.01821 [stat.ML].

[9] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*. 2016. arXiv: 1502.03044 [cs.LG].

[10] Thomas Dietterich, Richard Lathrop, and Tomás Lozano-Pérez. "Solving the Multiple Instance Problem with Axis-Parallel Rectangles". In: *Artificial Intelligence* 89 (Mar. 2001), pp. 31–71. DOI: 10.1016/S0004-3702(96)00034-3.

[11] Oded Maron and Tomás Lozano-Pérez. "A Framework for Multiple-Instance Learning". In: *Advances in Neural Information Processing Systems*. Ed. by M. Jordan, M. Kearns, and S. Solla. Vol. 10. MIT Press, 1997. URL: https://proceedings.neurips.cc/paper_files/paper/1997/file/82965d4ed8150294d4330ace00821d77-Paper.pdf.

[12] Gwenolé Quellec, Guy Cazuguel, Béatrice Cochener, and Mathieu Lamard. "Multiple-Instance Learning for Medical Image and Video Analysis". In: *IEEE Reviews in Biomedical Engineering* 10 (2017), pp. 213–234. DOI: 10.1109/RBME.2017.2651164.

[13] Maximilian Ilse, Jakub M. Tomczak, and Max Welling. *Attention-based Deep Multiple Instance Learning*. 2018. arXiv: 1802.04712 [cs.LG].

[14] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[15] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[16] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. 2019. arXiv: 1912.01703 [cs.LG].

[17] Anqi Mao, Mehryar Mohri, and Yutao Zhong. *Cross-Entropy Loss Functions: Theoretical Analysis and Applications*. 2023. arXiv: 2304.07288 [cs.LG].

[18] William Falcon and The PyTorch Lightning team. *PyTorch Lightning*. Version 1.8.6. Dec. 2022. DOI: 10.5281/zenodo.7469930. URL: https://doi.org/10.5281/zenodo.7469930.

[19] Gaspard Villa. *Exploring Intersectional Fairness in Histopathological Image Classifications*. 2024. URL: https://github.com/gaspardvilla/Intersectional-Fairness-on-Histopathological-images-classification.git.

[20] James R. Foulds; Rashidul Islam; Kamrun Naher Keya and Shimei Pan. "An intersectional definition of Fairness." In: *University of Maryland - Baltimore County - USA* (2019). DOI: https://doi.org/10.48550/arXiv.1807.08362.

[21] James R. Foulds; Rashidul Islam; Kamrun Naher Keya and Shimei Pan. "Bayesian Modeling of Intersectional Fairness - The Variance of Bias". In: (2020). DOI: https://doi.org/10.48550/arXiv.1811.07255.

[22] Emily Diana, Wesley Gill, Michael Kearns, Krishnaram Kenthapadi, and Aaron Roth. *Minimax Group Fairness: Algorithms and Experiments*. 2021. arXiv: 2011.03108 [cs.LG].

[23] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006. DOI: https://doi.org/10.1017/CBO9780511546921.

[24] Yoav Freund and Robert E. Schapire. "Game theory, on-line prediction and boosting". English (US). In: 1996, pp. 325–332. DOI: 10.1145/238061.238163.

[25] Arthur M Geoffrion. "Proper efficiency and the theory of vector maximization". In: *Journal of Mathematical Analysis and Applications* 22.3 (1968), pp. 618–630. ISSN: 0022-247X. DOI: https://doi.org/10.1016/0022-247X(68)90201-1.