MIME: Minority Inclusion for Majority Group Enhancement of AI Performance

Pradyumna Chari[1], Yunhao Ba[1], Shreeram Athreya[1], and Achuta Kadambi[1,2]

[1] Department of Electrical and Computer Engineering, UCLA
[2] Department of Computer Science, UCLA
{pradyumnac,yhba,shreeram}@ucla.edu, achuta@ee.ucla.edu

Supplementary Contents

This supplement is organized as follows:

- Section A contains the proof for Theorem 1.
- Section B contains the proof for Theorem 2.
- Section C contains the proof for Theorem 3.
- Section D discusses MIME existence beyond 1D.
- Section E describes further details for the feature space analysis.
- Section F contains implementation details across the six datasets.
- Section G describes additional secondary analysis.
- Section H describes our implementation of the hard mining comparison.
- Section I describes our code.
- Section J discusses potential negative ethical impacts of this work.

A Proof for Theorem 1

We consider the one-dimensional linear classifier setting, trained using the Perceptron algorithm. Given any $x \in \mathbb{R}$, the classifier evaluates an output y given by,

$$y = wx + b, \tag{1}$$

where $w, b \in \mathbb{R}$. The decision threshold in this case is at $y = 0$. For simplification, we reduce the redundant parameter, as follows:

$$y' = x + b'. \tag{2}$$

Note that the decision threshold is unaffected by this conversion. For notational simplicity, we use $y = y'$ and $b = b'$ here onward. We consider the perceptron decision and update rule, modified for our case. That is, for any training sample (x_i, y_i), the predicted output is given by,

$$\hat{y}_i = \frac{sign(x_i + b) + 3}{2}, \tag{3}$$

where $sign(\cdot)$ is the sign function. Readers will notice the unconventional form of this decision rule. The additional terms map the conventional perceptron labels in $\{-1, 1\}$ to our chosen labels $\{1, 2\}$ respectively.
For an appropriately chosen learning rate γ, the parameter update rule for this setting is given by:

$$b \leftarrow \begin{cases} b + \gamma, & \text{if } \hat{y}_i \neq y_i \text{ and } y_i = 2 \\ b - \gamma, & \text{if } \hat{y}_i \neq y_i \text{ and } y_i = 1 \end{cases}. \tag{4}$$

Let $\mathbf{h}_{\text{ideal}} \triangleq [1, b_{\text{ideal}}]^T$ denote the ideal decision hyperplane. Under the current assumption of no domain gap, it can be shown that this ideal hyperplane is located at $x = d_{\text{ideal}}$ such that,

$$\begin{aligned} p_1^{\text{minor}}(d_{\text{ideal}}) &= p_2^{\text{minor}}(d_{\text{ideal}}) \\ p_1^{\text{major}}(d_{\text{ideal}}) &= p_2^{\text{major}}(d_{\text{ideal}}). \end{aligned} \tag{5}$$

This also implies that $b_{\text{ideal}} = -d_{\text{ideal}}$. Now, consider an initial training set of $K-1$ samples from the majority group, $\mathcal{D}_{K-1}^{\text{major}}$. A decision hyperplane $\mathbf{h}_{K-1}$ is learnt from these samples. Then, without loss of generality, we can assume that,

$$d_{K-1} = d_{\text{ideal}} + \Delta. \tag{6}$$

That is, the real hyperplane $\mathbf{h}_{K-1}$ is non-ideally located closer to the positve class ($y = 2$) than $\mathbf{h}_{\text{ideal}}$. Δ is a small positive value representing the error in the learnt decision hyperplane. Consider that the K-th sample is drawn from the majority group x_K^{major}. Recall that parameter updates for the Perceptron algorithm take place only in the event of incorrect label estimation $\hat{y}_K \neq y_K$. If we denote the change in the parameter b due to this sample as Δb, then three cases exist:

1. Sample from class 2 is classified as belonging to class 1 such that $x_K^{\text{major}} \sim p_2^{\text{minor}}(x),\ x_K^{\text{major}} < d_{\text{ideal}} - \Delta$. Associated $\Delta b = +\gamma$.
2. Sample from class 2 is classified as belonging to class 1 such that $x_K^{\text{major}} \sim p_2^{\text{minor}}(x),\ d_{\text{ideal}} - \Delta \leq x_K^{\text{major}} < d_{\text{ideal}} + \Delta$. Associated $\Delta b = +\gamma$.
3. Sample from class 1 classified as belonging to class 2 such that $x_K^{\text{major}} \sim p_1^{\text{minor}}(x),\ x_K^{\text{major}} \geq d_{\text{ideal}} + \Delta$. Associated $\Delta b = -\gamma$.

Let the expected change in b due to one majority group sample be denoted as Δb^{major}. Δd^{major} is similarly defined for the expected change in d. Then, the following holds true:

$$\Delta b^{\text{major}} = \mathop{\mathbb{E}}_{x_K^{\text{major}}} [\Delta b]. \tag{7}$$

Writing out the expectation over all three cases,

$$\begin{aligned} \Delta b^{\text{major}} = \gamma \int_{x=-\infty}^{d_{\text{ideal}}-\Delta} p_2^{\text{major}}(x)dx + \gamma \int_{x=d_{\text{ideal}}-\Delta}^{d_{\text{ideal}}+\Delta} p_2^{\text{major}}(x)dx \\ - \gamma \int_{x=d_{\text{ideal}}+\Delta}^{+\infty} p_1^{\text{major}}(x)dx. \end{aligned} \tag{8}$$

Similar expressions can be identified if the K-th sample is drawn from the minority group. Under the assumption that the mixture models under consideration are symmetric Gaussian mixture models,

$$\int_{x=-\infty}^{d_{\text{ideal}}-\Delta} p_2^{\text{major}}(x)dx = \int_{x=d_{\text{ideal}}+\Delta}^{+\infty} p_1^{\text{major}}(x)dx. \tag{9}$$

Then, using Equation 8 and Equation 9,

$$\Delta b^{\text{major}} = \gamma \int_{x=d_{\text{ideal}}-\Delta}^{d_{\text{ideal}}+\Delta} p_2^{\text{major}}(x)dx. \tag{10}$$

The region between $x = d_{\text{ideal}} - \Delta$ and $d_{\text{ideal}} + \Delta$ determines the expected change in the classification parameter. If Δ is small enough, $\Delta b^{\text{major}} \approx 2\gamma p_2^{\text{major}}(d_{\text{ideal}})\Delta$. Similarly, $\Delta b^{\text{minor}} \approx 2\gamma p_2^{\text{minor}}(d_{\text{ideal}})\Delta$.
We now identify a sufficient condition where $p_2^{\text{minor}}(x) > p_2^{\text{major}}(x)$ for $-\Delta \leq x \leq \Delta$, given that the overlaps satisfy the condition $O_{\text{minor}} > O_{\text{major}}$, as defined in the main text. Under the GMM assumption,

$$p_2^{\text{major}}(x) = \frac{1}{\sqrt{2\pi(\sigma_2^{\text{major}})^2}} exp\left(-\frac{(x-\mu_2^{\text{major}})^2}{2(\sigma_2^{\text{major}})^2}\right). \tag{11}$$

A similar expression exists for the minority group distribution as well. We wish to find the intersection point for the majority and minority distributions, that is $p_2^{\text{major}}(x) = p_1^{\text{major}}(x)$ for some x. This expression reduces to,

$$\frac{\left(x-\mu_2^{\text{major}}\right)^2}{{\sigma^{\text{major}}}^2} - \frac{\left(x-\mu_2^{\text{minor}}\right)^2}{{\sigma^{\text{minor}}}^2} = 2ln\left|\frac{\sigma^{\text{minor}}}{\sigma^{\text{major}}}\right|. \tag{12}$$

We want to ensure that this intersection point occurs for an $x > d_{\text{ideal}}$. This sets up a hyperbolic equation for the condition. For our purposes of proving existence, we qualitatively note that if the majority group variance is not very large (meaning the likelihood of sampling at the ideal hyperplane is low for the majority group), and the minority group variance is not very large (such that it does not tend close to a uniform distribution), $p_2^{\text{minor}}(x) > p_2^{\text{major}}(x)$. Then,

$$\Delta b^{\text{minor}} > \Delta b^{\text{major}}. \tag{13}$$

$$\Delta d^{\text{minor}} < \Delta d^{\text{major}} < 0. \tag{14}$$

Our final task is to relate the expected change in the decision hyperplane over a choice of training sets $\mathcal{D}_K^+$ and $\mathcal{D}_K^-$, with associated learnt hyperplanes $\mathbf{h}_K^+$ and $\mathbf{h}_K^-$. As a reminder,

$$\begin{aligned} \mathcal{D}_K^+ &= \{\mathcal{D}_{K-1}^{\text{major}}, x_K^{\text{major}}\} \\ \mathcal{D}_K^- &= \{\mathcal{D}_{K-1}^{\text{major}}, x_K^{\text{minor}}\}, \end{aligned} \tag{15}$$

Consider a general training setting, where we use minibatches of size $M > 1$, over multiple epochs. Then, any minibatch containing the K-th sample can be split into the K-th sample and a random subset of $M-1$ samples from $\mathcal{D}_{K-1}^{\text{major}}$. Therefore, on average, the only difference to the sample updates would be due to the contributions of the K-th sample. This brings us to our final observations,

$$\begin{aligned} \mathop{\mathbb{E}}_{x_K^{\text{minor}}}[d_K^+] &= d_{K-1}^{\text{major}} + \Delta d^{\text{major}} \\ \mathop{\mathbb{E}}_{x_K^{\text{minor}}}[d_K^-] &= d_{K-1}^{\text{major}} + \Delta d^{\text{minor}}. \end{aligned} \tag{16}$$

From Equations 14 and 16,

$$\mathop{\mathbb{E}}_{x_K^{\text{minor}}}[d_K^-] < \mathop{\mathbb{E}}_{x_K^{\text{minor}}}[d_K^+], \text{ and} \tag{17}$$

$$\mathop{\mathbb{E}}_{x_K^{\text{minor}}}[|d_{\text{ideal}} - d_K^-|] < \mathop{\mathbb{E}}_{x_K^{\text{minor}}}[|d_{\text{ideal}} - d_K^+|]. \tag{18}$$

The above holds for small enough γ. Since we know the relationship between the decision hyperplane $\mathbf{h}$ and the associated d in our setting, the following equations hold true:

$$\mathop{\mathbb{E}}_{x_K^{\text{minor}}}\|\mathbf{h}_{\text{ideal}} - \mathbf{h}_K^-\| < \mathop{\mathbb{E}}_{x_K^{\text{minor}}}\|\mathbf{h}_{\text{ideal}} - \mathbf{h}_K^+\|, \tag{19}$$

$$\mathop{\mathbb{E}}_{x_K^{\text{minor}}}\mathcal{P}^{\text{major}}\{\mathbf{h}_K^-\} < \mathop{\mathbb{E}}_{x_K^{\text{major}}}\mathcal{P}^{\text{major}}\{\mathbf{h}_K^+\}. \blacksquare \tag{20}$$

B Proof for Theorem 2

We follow a similar approach as in Theorem 1. Let $\mathbf{h}_{\text{ideal}}^{\text{major}} \triangleq [1, b_{\text{ideal}}^{\text{major}}]^T$ denote the ideal decision hyperplane for the majority group. Let $\mathbf{h}_{\text{ideal}}^{\text{minor}} \triangleq [1, b_{\text{ideal}}^{\text{minor}}]^T$ denote the ideal decision hyperplane for the minority group. Then, the ideal hyperplanes are located at $x = d_{\text{ideal}}^{\text{major}}$ and $x = d_{\text{ideal}}^{\text{minor}}$ respectively such that,

$$\begin{aligned} p_1^{\text{minor}}(d_{\text{ideal}}^{\text{minor}}) &= p_2^{\text{minor}}(d_{\text{ideal}}^{\text{minor}}) \\ p_1^{\text{major}}(d_{\text{ideal}}^{\text{major}}) &= p_2^{\text{major}}(d_{\text{ideal}}^{\text{major}}). \end{aligned} \tag{21}$$

This implies that $b_{\text{ideal}}^{\text{major}} = -d_{\text{ideal}}^{\text{major}}$ and $b_{\text{ideal}}^{\text{minor}} = -d_{\text{ideal}}^{\text{minor}}$. Consider an initial training set of $K-1$ samples from the majority group, $\mathcal{D}_{K-1}^{\text{major}}$. Then, without loss of generality, we can assume that $d_{K-1} = d_{\text{ideal}}^{\text{major}} + \Delta$, where $\Delta > 0$. Additionally, we consider the existence of domain gap in this case, that is, $d_{\text{ideal}}^{\text{minor}} = d_{\text{ideal}}^{\text{major}} + \delta$. Let $\delta < \Delta$. Similar to the setting in Theorem 1 (Equation 8), we can set up the equation for expected parameter change in the case of the majority and minority

groups as follows:

$$\Delta b^{\text{major}} = \gamma \int_{x=-\infty}^{d_{\text{ideal}}^{\text{major}}-\Delta} p_2^{\text{major}}(x)dx + \gamma \int_{x=d_{\text{ideal}}^{\text{major}}-\Delta}^{d_{\text{ideal}}^{\text{major}}+\Delta} p_2^{\text{major}}(x)dx - \gamma \int_{x=d_{\text{ideal}}^{\text{major}}+\Delta}^{+\infty} p_1^{\text{major}}(x)dx. \quad (22)$$

$$\Delta b^{\text{minor}} = \gamma \int_{x=-\infty}^{d_{\text{ideal}}^{\text{minor}}-(\Delta-\delta)} p_2^{\text{minor}}(x)dx + \gamma \int_{x=d_{\text{ideal}}^{\text{minor}}-(\Delta-\delta)}^{d_{\text{ideal}}^{\text{minor}}+(\Delta-\delta)} p_2^{\text{minor}}(x)dx - \gamma \int_{x=d_{\text{ideal}}^{\text{minor}}+(\Delta-\delta)}^{+\infty} p_1^{\text{minor}}(x)dx. \quad (23)$$

Under the assumption that the mixture models under consideration are symmetric Gaussian mixture models,

$$\Delta b^{\text{major}} = \gamma \int_{x=d_{\text{ideal}}^{\text{major}}-\Delta}^{d_{\text{ideal}}^{\text{major}}+\Delta} p_2^{\text{major}}(x)dx, \quad (24)$$

$$\Delta b^{\text{minor}} = \gamma \int_{x=d_{\text{ideal}}^{\text{minor}}-(\Delta-\delta)}^{d_{\text{ideal}}^{\text{minor}}+(\Delta-\delta)} p_2^{\text{minor}}(x)dx. \quad (25)$$

If $\Delta + |\delta|$ is small enough,

$$\Delta b^{\text{major}} \approx 2\gamma p_2^{\text{major}}(d_{\text{ideal}}^{\text{major}})\Delta, \quad (26)$$

$$\Delta b^{\text{minor}} \approx 2\gamma p_2^{\text{minor}}(d_{\text{ideal}}^{\text{minor}})(\Delta - \delta). \quad (27)$$

By establishing the same conditions on group class variances as Theorem 1, we know that $p_2^{\text{minor}}(d_{\text{ideal}}^{\text{minor}}) > p_2^{\text{major}}(d_{\text{ideal}}^{\text{major}})$. We now identify conditions under which $\Delta b^{\text{minor}} > \Delta b^{\text{major}}$.

Case 1 - $\boldsymbol{\delta < 0}$*:* Under the same conditions as Theorem 1, $(\Delta - \delta) > \Delta$, and $p_2^{\text{minor}}(d_{\text{ideal}}^{\text{minor}}) > p_2^{\text{major}}(d_{\text{ideal}}^{\text{major}})$. Therefore,

$$\Delta b^{\text{minor}} > \Delta b^{\text{major}}. \quad (28)$$

Case 2 - $\boldsymbol{\delta > 0}$*:*

$$\Delta b^{\text{major}} \approx 2\gamma p_2^{\text{major}}(d_{\text{ideal}}^{\text{major}})\Delta, \quad (29)$$

$$\Delta b^{\text{minor}} \approx 2\gamma p_2^{\text{minor}}(d_{\text{ideal}}^{\text{minor}})(\Delta - \delta). \quad (30)$$

For $\Delta b^{\text{minor}} > \Delta b^{\text{major}}$,

$$p_2^{\text{minor}}(d_{\text{ideal}}^{\text{minor}})(\Delta - \delta) > p_2^{\text{major}}(d_{\text{ideal}}^{\text{major}})\Delta. \quad (31)$$

Rearranging Equation 31,

$$\frac{p_2^{\text{major}}(d_{\text{ideal}}^{\text{major}})}{p_2^{\text{minor}}(d_{\text{ideal}}^{\text{minor}})} < \left(1 - \frac{\delta}{\Delta}\right). \tag{32}$$

Given the definitions of the majority and minority groups,

$$p_2^{\text{major}}(d_{\text{ideal}}^{\text{major}}) < p_2^{\text{minor}}(d_{\text{ideal}}^{\text{minor}}), \tag{33}$$

$$O_{\text{major}} < O_{\text{minor}}. \tag{34}$$

Since all four of these terms depend only on the means and variances of the Gaussian components, we can write,

$$\frac{O_{\text{major}}}{O_{\text{minor}}} = \frac{p_2^{\text{major}}(d_{\text{ideal}}^{\text{major}})}{p_2^{\text{minor}}(d_{\text{ideal}}^{\text{minor}})} f, \tag{35}$$

where f is a positive scalar constant that depends only on the component means and variances. From Equations 32 and 35,

$$\frac{O_{\text{major}}}{O_{\text{minor}}} < \left(1 - \frac{\delta}{\Delta}\right) f. \tag{36}$$

This proves the conditions in the theorem. Theorem 1 can now be used to show the existence of the MIME effect in the presence of domain gap, for these conditions. ■

A Note on the Theorems: Theorems 1 and 2 are existence theorems. That is, they show that there exist certain conditions under which the MIME effect can be observed. The theorems make these arguments based on the 'usefulness' of points close to the ideal hyperplane. The direct metric of correlation is the likelihood for a particular distribution to sample at the ideal hyperplane. However, since this cannot be easily measured in practice, we set up our proofs in terms of a correlated metric: the overlap.

C Proof for Theorem 3

This Theorem considers distributions with general prior distributions. Therefore, for the majority group, let

$$\begin{aligned} p_2^{\text{major}'}(x) &= \pi^{\text{major}} p_2^{\text{major}}(x), \\ p_q^{\text{major}'}(x) &= (1 - \pi^{\text{major}}) p_1^{\text{major}}(x). \end{aligned} \tag{37}$$

Similar definitions are made for the minority group as well. Then, assuming $d_{K-1} = d_{\text{ideal}}^{\text{major}} + \Delta$, $\Delta > 0$ (similar to Theorem 2), and $\delta = 0$ (for now), and

drawing from Equation 8), we can set up the equation for expected parameter change in the case of the majority group as follows:

$$\begin{aligned}\Delta b^{\text{major}} &= \gamma \int_{x=-\infty}^{d_{\text{ideal}}+\Delta} p_2^{\text{major}'}(x)dx - \gamma \int_{x=d_{\text{ideal}}+\Delta}^{+\infty} p_1^{\text{major}'}(x)dx \\ &= T^{\text{major}}(d_{\text{ideal}} + \Delta).\end{aligned} \tag{38}$$

A similar expression holds true for the minority group. Then, if $T^{\text{major}}(d_{\text{ideal}} + \Delta) < T^{\text{minor}}(d_{\text{ideal}} + \Delta)$, the MIME effect will hold true.

Similarly, if $d_{K-1} = d_{\text{ideal}}^{\text{major}} - \Delta$, $\Delta > 0$,

$$\begin{aligned}\Delta b^{\text{major}} &= -\gamma \int_{x=-\infty}^{d_{\text{ideal}}-\Delta} p_2^{\text{major}'}(x)dx + \gamma \int_{x=d_{\text{ideal}}-\Delta}^{+\infty} p_1^{\text{major}'}(x)dx \\ &= -T^{\text{major}}(d_{\text{ideal}} - \Delta).\end{aligned} \tag{39}$$

Then, if $-T^{\text{major}}(d_{\text{ideal}} - \Delta) < -T^{\text{minor}}(d_{\text{ideal}} - \Delta)$, the MIME effect will hold true.

Combining the two expressions, for a sufficient existence condition, we get,

$$\begin{aligned}\min\left\{T^{\text{minor}}(d_{\text{ideal}} + \Delta), -T^{\text{minor}}(d_{\text{ideal}} - \Delta)\right\} > \\ \max\left\{T^{\text{major}}(d_{\text{ideal}} + \Delta), -T^{\text{major}}(d_{\text{ideal}} - \Delta)\right\}.\end{aligned} \tag{40}$$

This completes the proof. ■

Note that the existence proof for Theorem 3 ignores the effect of domain gap δ, in the interest of readability and brevity. A very similar existence proof can be established with domain gap. We omit the derivation and provide the final condition below (under the constraints on δ and Δ as in Theorem 2, and using the same notation):

$$\begin{aligned}\min\left\{T^{\text{minor}}(d_{\text{ideal}}^{\text{major}} + \Delta), -T^{\text{minor}}(d_{\text{ideal}}^{\text{major}} - \Delta)\right\} > \\ \max\left\{T^{\text{major}}(d_{\text{ideal}}^{\text{major}} + \Delta), -T^{\text{major}}(d_{\text{ideal}}^{\text{major}} - \Delta)\right\}.\end{aligned} \tag{41}$$

D MIME Existence Beyond 1D Settings

Consider $\mathbf{x} \in \mathbb{R}^n$. The perceptron decisions are based on the metric $y = \mathbf{w}^T\mathbf{x} + b$, where $\mathbf{w} \in \mathbb{R}^n$, and $y, b \in \mathbb{R}$. Similar to Theorem 1, we consider the perceptron decision and update rule. That is, for any training sample $(\mathbf{x}_i, y_i)$, the predicted label is given by,

$$\hat{y}_i = \frac{sign(\mathbf{w}^T\mathbf{x}_i + b) + 3}{2}. \tag{42}$$

We can rewrite this in terms of a single decision hyperplane by defining $\tilde{\mathbf{w}} = [\mathbf{w}^T\ b]^T$ and $\tilde{\mathbf{x}} = [\mathbf{x}^T\ 1]^T$. For a small learning rate γ, the updated decision rule becomes,

$$\hat{\tilde{y}}_i = \frac{sign(\tilde{\mathbf{w}}^T \tilde{\mathbf{x}}_i) + 3}{2}. \tag{43}$$

$$\tilde{\mathbf{w}} \leftarrow \begin{cases} \tilde{\mathbf{w}} + \gamma \tilde{\mathbf{x}}_i, & \text{if } \hat{y}_i \neq y_i \text{ and } y_i = 2 \\ \tilde{\mathbf{w}} - \gamma \tilde{\mathbf{x}}_i, & \text{if } \hat{y}_i \neq y_i \text{ and } y_i = 1 \end{cases}. \tag{44}$$

We now refer to the hyperplane $\tilde{\mathbf{w}}$ as the decision hyperplane. Let $\mathbf{h}_{\text{ideal}}$ be the ideal decision hyperplane. In this setting, any domain gap δ or error in real hyperplane estimation Δ manifests as a direction/angle error in the hyperplane normal vector (since the bias term b is subsumed in the hyperplane). The updates change the normal vector of the hyperplane through a linear combination with the sample $\tilde{\mathbf{x}}_i$, scaled by the learning rate γ.

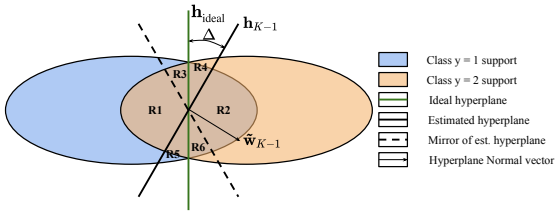

Fig. A. The MIME effect holds in a multidimensional setting as well. We show the support for the two finite distributions. Weight vector updates arising out of samples from regions R3, R4, R5 and R6 lead to an update with a large vertical (corrective) component (favorable update). Updates arising out of regions R1 and R2 result in an overall update in the horizontal direction (unfavorable update).

We now provide a qualitative description for the existence of the MIME effect, in terms of the likelihood of a favorable update to $\tilde{\mathbf{w}}$. We consider a simplified 2D case with symmetric distributions and $\delta = 0$. A finite support is assumed for the majority and minority groups, for ease of understanding. Consider that the bias term b is known, and only the hyperplane direction is to be refined. Again, we denote the hyperplane from our finite training set $\mathcal{D}_{K-1}^{\text{major}}$ as $\mathbf{h}_{K-1}$. The error Δ in this case is now the angular error between the normals for $\mathbf{h}_{\text{ideal}}$ and $\mathbf{h}_{K-1}$. Figure A indicates this setting. The learnt hyperplane $\tilde{\mathbf{w}}_{K-1}$ is shown as a black solid line. The black dashed line represents the mirror image of the learnt hyperplane, defined for aid in simplification. Recall that updates to the weight vector take place on misclassification. On average, the updates due to samples in regions R1 for $(y = 2)$ and R2 (for $y = 1$) lead to a net horizontal (leftward) weight update. This is an unfavorable update that increases Δ. Therefore, the favorable updates on average are from regions R3 and R4 for $y = 2$, and R5 and R6 for $y = 1$. This is a net update with large vertical (upward) update. This is

a favorable update that decreases Δ. These regions are described based on the small angular deviation Δ. Since the distributions have finite support along the direction parallel to the ideal hyperplane (vertical direction in Figure A), the requirement again reduces to greater likelihood of sampling close to the ideal hyperplane (similar to Theorems 1 and 2), since Δ is small. That is, distributions that sample close to the ideal hyperplane with greater probability have a greater expected likelihood of a favorable update. Under similar conditions as Theorem 1, MIME effect holds in this case.
The extension to include the bias term b is straightforward. We follow the setting in Equation 43 and subsume the bias as part of the weights. In this case, Δ includes the error in both the hyperplane normal direction as well as the bias. Extensions to greater number of dimensions can be done using the same arguments. Additionally, domain gap can also be introduced. We omit explicit mathematical expressions in the interest of brevity, and since our goal here is to establish existence.

E Feature Space Analysis

Constructing the Projected Feature Histograms: Let f denote a feature vector, in the penultimate layer of a classification neural network. For example, in the case of ResNet-34 [7], $\mathbf{f} \in \mathbb{R}^{512}$. Similarly, let $\mathbf{w}$ be the final layer weights. In the case of multiple final layer hyperplanes, we choose any one of the hyperplanes (since for the two class classification task, the two projected variables are correlated when trained against the cross entropy loss for 2 classes). Then, we define $x \in \mathbb{R}$ as,

$$x = \mathbf{w}^T \mathbf{f}. \tag{45}$$

Classification decisions are made solely on the basis of the projected variable x. Therefore, we analyze the histogram distributions for x. Practically, for each dataset, we use the best performing (in terms of majority group performance) model trained using a minority training fraction (β) of 0.5. This is chosen in order to obtain histograms of x for all four distributions – the two task classes for both the majority and minority groups. The histograms are created using the test set samples.

Estimating the Overlap: The overlap is estimated from the histograms, using the following Python code snippet:

```
def histogram_intersection(h1, h2, bins):
    #INPUTS:
    #h1, h2: normalized histograms
    #bins: number of bins in the histograms (should be
                                    equal for the two
                                    histograms)
    #OUTPUTS:
    #sm: overlap fraction
    sm = 0
```

```
    for i in range(bins):
        sm += min(h1[i], h2[i])
    return sm
```

Estimating the Domain Gap: We follow a two step process to estimate the domain gap δ. First, the ideal decision hyperplanes for the majority and minority groups are estimated, using Equation 21. We fit a fifth order polynomial to the two histograms. The central intersection point of the histograms (i.e the intersection point that lies between the means of the two classes) is then the location of the ideal decision threshold. The following Python code snippet describes this:

```
import numpy as np

def ideal_hyperplane(h1, h2, z, ref=5):
    #INPUTS:
    #h1, h2: the two histograms, of equal length and
                                    identical bins
    #z: a list of the histogram bin centers
    #ref: Search space for the intersection of the two
                                    histograms- default is
                                     from -5 to 5
    #OUTPUT:
    #z_dec: Ideal decision threshold between the two
                                    histogram
                                    distributions
    z_dash = np.polyfit(z, h1, 5)
    f1 = np.poly1d(z_dash)
    # calculate polynomial
    z_dash = np.polyfit(z, f2, 5)
    f2 = np.poly1d(z_dash)
    new_z = np.linspace(-ref,ref,5000)
    new_f1 = f1(new_z)
    new_f2 = f2(new_z)
    id_dec = np.argmin(np.abs(new_f1-new_f2))
    z_dec = new_z[id_dec]
    return z_dec
```

The domain gap is the absolute difference between two ideal decision thresholds, for each of the two group classes. Figure 3 of the main paper may be referred to for a graphical visualization.

Notes on the Estimated Measures: The latent feature space analysis is not perfect. This is because the feature extraction part of the network is jointly learnt along with decision hyperplane. Histograms are plotted on the 50% minority training ratio so as to enable a fair domain gap and overlap comparison between the two group classes. Specifically, note that we define task complexity in the main paper in terms of the minority only and majority only train sets which deviates from the setting here. The estimates for overlap and domain gap are therefore approximate correlated estimates and not exact measures.

Analysis of Feature Space Gaussian-like Behavior: We set up the Chi-Squared goodness of fit test on all 20 distributions under consideration (i.e. across 5 datasets and 4 distributions each per dataset). These statistics correspond to the distributions in Table 1 and Figure 4 of the main paper. Python code for testing the hypotheses is given below. The number of bins are chosen so as to ensure ≥ 5 samples per bin on average.

```
from scipy.stats import chisquare
from scipy.stats import norm
from scipy import stats
import pandas as pd

def chi_square_stats(vals,no_bins)
    #INPUTS:
    #vals: a list of samples whose Gaussianity is to be
                                    tested
    #no_bins: number of bins (thumb rule: no_bins<len(
                                    vals)/5)

    tot_vals = len(vals)
    # mean and standard deviation of given data
    mean = np.mean(vals)
    std = np.std(vals)

    interval = []
    for i in range(1,no_bins+1):
      val = stats.norm.ppf(i/no_bins, mean, std)
      interval.append(val)
    interval.insert(0, -np.inf)

    lower = interval[:-1]
    upper = interval[1:]

    df = pd.DataFrame({'lower_limit':lower, 'upper_limit'
                                    :upper})


    sorted_vals = list(sorted(vals))
    df['obs_freq'] = df.apply(lambda x:sum([i>x['
                                    lower_limit'] and i<=x
                                    ['upper_limit'] for i
                                    in sorted_vals]), axis
                                    =1)
    df['exp_freq'] = tot_vals/no_bins

    statistic = stats.chisquare(df['obs_freq'], df['
                                    exp_freq'])

    p = 2    # number of parameters for 1D Gaussian
    DOF = len(df['obs_freq']) - p -1
```

```
    thresh = stats.chi2.ppf(0.95, DOF)

    return statistic, thresh
```

Table A highlights the evaluated chi-square statistics, as well as related parameters. Note that a lower value of the statistic is better, and the null hypothesis is not rejected when the value of the statistic is lower than the critical value. We establish the null hypothesis at a 5% level of significance for each distribution to be that the samples are drawn from a Gaussian distribution. Distributions that are unable to reject the null hypothesis are indicated in bold. It can be seen that a large majority of the distributions indicate that the projected latent features follow a Gaussian-like distribution.

Table A. Chi-Squared goodness of fit measures for all distributions. Distributions with bolded values show the estimated statistics that are lower than the critical value, indicating that the null hypothesis (Gaussian distribution) cannot be rejected.

Dataset	No. of samples per group per class	No. of Bins	Critical Value	Majority Group		Minority Group	
				$y=1$	$y=2$	$y=1$	$y=2$
DS-1 [10]	379	15	21.03	**13.65**	28.69	**10.25**	**15.39**
DS-2 [5]	126	15	21.03	**7.81**	**12.10**	**11.62**	**9.24**
DS-4 [16]	126	15	21.03	**10.43**	**17.57**	**17.10**	**4.48**
DS-5 [1]	159	15	21.03	**11.09**	24.05	**5.74**	**14.40**
DS-6 [18, 19]	43	5	5.99	25.48	**5.02**	**5.72**	**4.79**

F Implementation Details

Analysis measures: For each task, we estimate the test accuracy $a_p^i(\beta)$ as a function of minority group fraction in the train set $\beta \in [0,1]$, for a trial $i \in \{1,...,N\}$, for a group class g (e.g. dark skin tones). N is the total number of trials. Practically, we evaluate performance for a finite set of β values, represented by the set $B = \{0, 0.1, 0.2, \ldots, 1.0\}$. We now define the following measures.
Average accuracy: For a given minority training ratio β_0, and for a given group class g, we define the average accuracy,

$$\bar{a}_g(\beta_0) = \frac{1}{N}\sum_{i=1}^{N} a_g^i(\beta_0). \tag{46}$$

Error bounds: We also evaluate the *trend variation* among $a_g^i(\beta)$ for various i. That is, we want to evaluate if across all the trials (for a particular task-dataset combination), the relative trend (of majority group performance gain) holds true. One candidate measure for this is $std_i(a_g^i(\beta))$ for each β, where $std_i(\cdot)$ is the standard deviation operator, over i. However, this measure will include average changes in accuracy for all splits, for a particular trial (arising out of unrelated effects such as different train or test set samples). This is unnecessary

in our case. Therefore, we define our error measure $\hat{\zeta}(\beta)$ as the β-mean subtracted standard deviation. That is,

$$\begin{aligned} \hat{\zeta}(\beta) &= std_i(a_g^i(\beta) - \bar{a}_g^i), \\ \bar{a}_g^i &= \frac{1}{|B|}\sum_{\beta \in B} a_g^i(\beta), \end{aligned} \tag{47}$$

where $|\cdot|$ is the cardinality operator representing the size of a set. In our graphs, we plot the average accuracy $\bar{a}_g(\beta)$ as well as the error bounds, from $\bar{a}_g(\beta) - \hat{\zeta}(\beta)$ to $\bar{a}_g(\beta) + \hat{\zeta}(\beta)$, $\forall \beta \in B$.

Network Architectures Used: For all the vision-related experiments, we use the ResNet-34 architecture [7]. We only modify the output layer of the network so as to match the number of task classes (9 for Dataset 3, and 2 for all other tasks). For the Adult (Census) Dataset [1], we use a fully connected network with sigmoid outputs. The PyTorch [15] implementation for the model is included below.

```
#Model

def act(x):
    return F.relu(x)

class Network(nn.Module):
    def __init__(self,):
        super().__init__()
        self.fc1 = nn.Linear(101, 50)
        self.fc2 = nn.Linear(50, 50)
        self.fc3 = nn.Linear(50, 50)
        self.fcLast = nn.Linear(50,2)

    def forward(self,x):

        x = act(self.fc1(x))
        # x = self.b1(x)
        x = act(self.fc2(x))
        x = act(self.fc3(x))
        x = torch.sigmoid(self.fcLast(x))
        return x
```

General Experiment Details: All experiments were carried out using PyTorch [15]. Table B highlights the training parameters used for each dataset. We use different parameters for each of the datasets. These are experimentally chosen to maximize accuracy. All the models are trained using the AdamW optimizer [13] and the cross entropy loss. The train and test set sizes vary slightly across trials, due to different data splits. However, the train set size remains the same

Table B. Training configuration and parameters for all datasets and experiments. Parameters for each dataset are chosen so as to maximize performance.

Dataset (Task)	DS-1 [10] (Gender)	DS-2 [5] (Species)	DS-3 [19] (Age)	DS-4 [16] (Diagnosis)	DS-5 [1] (Income)	DS-6 [18, 19] (Gender)
Group class	Race	Skin tone	Gender	Gender	Gender	Species
Train set size	10900	1500	7700	1500	2600	750
Test set size (per group)	760	250	970	250	300	90
No. of trials	5	5	5	7	5	5
No. of epochs	35	60	65	40	250	20
Learning rate	0.0005	0.0006	0.0006	0.0006	0.0005	0.0005
Weight Decay	0.08	0.05	0.05	0.05	0.08	0.08
Input Shape/Config.	3x100x100	3x256x256	3x100x100	3x256x256	101x1	3x100x100

for all minority training ratios of a particular trial. A validation set is held out but given the small sample size of several datasets, we measure trends based on best test performance. This is to minimize the effect of sample specific performance gap in small datasets. Averaging of trends over multiple trials, and hence multiple train-test splits ensures that the trends do not overfit to a particular configuration. Each trial is run using a unique random seed. Table C highlights the random seeds used for our experiments, which were randomly chosen. Input images are resized to the chosen input size for each dataset. For the Adult dataset [1], we use a one-hot encoding scheme for the input. The group class information is dropped from the input before passing to the network. For all the datasets, across all minority training ratios for a particular trial, we use a fixed model initialization to ensure that the changes in accuracy are completely attributable to the train data configuration.

Table C. Random seeds used for the trials. Seeds were chosen at random for trials to generate average trends and error bounds.

Dataset (Task)	DS-1 [10] (Gender)	DS-2 [5] (Species)	DS-3 [19] (Age)	DS-4 [16] (Diagnosis)	DS-5 [1] (Income)	DS-6 [18, 19] (Gender)
Random Seeds	0, 1, 3, 5, 7	21, 42, 35, 28, 31	0, 55, 2, 15, 6	33, 42, 24, 36 54, 21, 28	13, 15, 17, 19, 21	0, 1, 3, 5, 9

Dataset Specific Information: To perform experiments on the **Pet Images Dataset**, we manually annotate light and dark fur cats and dogs from the larger dataset used in [5]. For the age classification task on the **UTKFace Dataset** [19], we pre-process the age labels to match the annotation format for the FairFace dataset [10]. For the large domain gap gender classification task using the **UTKFace and Chicken Images Datasets** [19, 18], we perform gender classification over human and chicken groups. Therefore, this experiment is over a new, composite dataset.

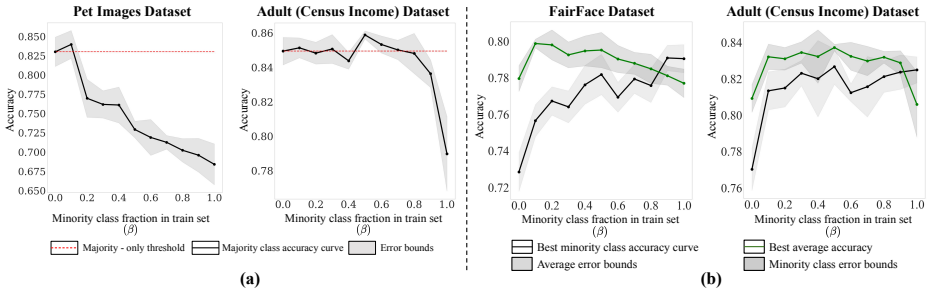

Fig. B. The MIME effect is complementary to data debiasing methods and consistent with research aimed at equal representation (ER) datasets. (a) Training configurations using data debiasing methods [3] show the MIME effect. (b) While ER datasets are not optimal for the MIME effect (Figure 5 and 6, main paper), optimal overall performance is observed close to ER.

G Additional Secondary Analysis of MIME

MIME effect with debiasing methods: We now analyze the interaction of the MIME effect with existing debiasing methods. Specifically, while applying hard-sample mining [3] (as an exemplary case) across the task classes ($y = 1, 2$), we sweep across various minority training ratios. Figure B(a) shows results on two datasets (implementation details may be found in the following section). The MIME effect continues to be observed. Debiasing methods act on the task classes ($y = 1, 2$) in an effort to improve performance while MIME acts on majority and minority groups, regardless of the task class. Therefore, MIME is complementary to debiasing methods, rather than a competitor. In our experiments, hard-sample mining does not lead to significant performance gains since the task classes are balanced by experimental design. In other scenarios where this might not be the case, MIME and hard sample mining might together improve performance.
Reconciling MIME with existing equal representation (ER) datasets: In this paper, we focus only on majority group performance, for which ER training datasets are not optimal in general. In contrast, existing efforts [4, 2, 11, 17, 12, 14, 8, 6, 9] focus on ER datasets to maximize overall (majority+minority) performance. This need not be optimal but is a good thumb rule. This is because while majority group performance eventually reduces with minority training ratio, minority group performance increases (Figure B(b) highlights this).

H Hard Mining Baseline Implementation

We implement a version of the method proposed in [3]. From a batch of 30 samples, 12 samples (6 of each task class) are retained and used in the training step. These are the samples with least confidence, with respect to ground truth targets. Code is shown below. Trial random seeds are the same as shown in Table C.

```
class compute_crossentropyloss_hardMine:
    """
    y0 is the vector with shape (batch_size,C)
    x shape is the same (batch_size), whose entries are
                                integers from 0 to C-1
    In our case, C=2.
    """
    def __init__(self, ignore_index=-100) -> None:
        self.ignore_index=ignore_index

    def __call__(self, y0, x):
        loss = 0.
        eps = 1e-5
        K = 6
        n_batch, n_class = y0.shape
        pos_score = torch.ones(n_batch).to(device)
        neg_score = torch.ones(n_batch).to(device)
        ix_pos = 0
        ix_neg = 0
        for y1, x1 in zip(y0, x):
            class_index = int(x1.item())
            score = torch.exp(y1[class_index])/(torch.exp(y1)
                                            .sum()+eps)
            if class_index == 0:
                neg_score[ix_neg] = score
                ix_neg+=1
            else:
                pos_score[ix_neg] = score
                ix_pos+=1

        pos_score,_ = torch.sort(pos_score,dim=0)
        neg_score,_ = torch.sort(neg_score,dim=0)
        pos_els = np.minimum(K,ix_pos)
        neg_els = np.minimum(K,ix_neg)
        for ix in np.arange(pos_els):
            loss = loss -torch.log(pos_score[ix])

        for ix in np.arange(neg_els):
            loss = loss -torch.log(neg_score[ix])

        loss = loss/(pos_els+neg_els)
        torch.cuda.empty_cache()
        return loss
```

I Our Code

Our code may be accessed through the project webpage at `https://visual.ee.ucla.edu/mime.htm/`. We provide code and guidance to perform experiments

on all six datasets. Due to specific requirements for each dataset, we provide six Jupyter notebooks. We also include details on setting up file structures and link to datasets wherever necessary. Please refer to the README file for further details.

J Negative Impacts and Mitigation

This paper focuses on highlighting the existence of the MIME effect, and not optimal configurations for performance gain. Nevertheless, potential negative outcomes may occur if the results are misinterpreted as guidance on dataset construction with respect to certain stakeholder groups. The rigor of our theoretical results emphasizes this nuance to computer scientists, and future work in diverse venues can extend the notion of minority inclusion for majority group performance gains to broader audiences.

References

1. Blake, C.L., Merz, C.J.: Uci repository of machine learning databases, 1998 (1998)
2. Buolamwini, J., Gebru, T.: Gender shades: Intersectional accuracy disparities in commercial gender classification. In: Conference on fairness, accountability and transparency. pp. 77–91. PMLR (2018)
3. Dong, Q., Gong, S., Zhu, X.: Class rectification hard mining for imbalanced deep learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1851–1860 (2017)
4. Gebru, T., Morgenstern, J., Vecchione, B., Vaughan, J.W., Wallach, H., Daumé III, H., Crawford, K.: Datasheets for datasets. arXiv preprint arXiv:1803.09010 (2018)
5. Golle, P.: Machine learning attacks against the asirra captcha. In: Proceedings of the 15th ACM conference on Computer and communications security. pp. 535–542 (2008)
6. Gong, Z., Zhong, P., Hu, W.: Diversity in machine learning. IEEE Access **7**, 64323–64350 (2019)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
8. Jo, E.S., Gebru, T.: Lessons from archives: Strategies for collecting sociocultural data in machine learning. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency. pp. 306–316 (2020)
9. Kadambi, A.: Achieving fairness in medical devices. Science **372**(6537), 30–31 (2021)
10. Karkkainen, K., Joo, J.: Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1548–1558 (2021)
11. Larrazabal, A.J., Nieto, N., Peterson, V., Milone, D.H., Ferrante, E.: Gender imbalance in medical imaging datasets produces biased classifiers for computer-aided diagnosis. Proceedings of the National Academy of Sciences **117**(23), 12592–12594 (2020)
12. Li, Y., Vasconcelos, N.: Repair: Removing representation bias by dataset resampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9572–9581 (2019)
13. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
14. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. ACM Computing Surveys (CSUR) **54**(6), 1–35 (2021)
15. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
16. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., et al.: Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. arXiv preprint arXiv:1711.05225 (2017)
17. Ryu, H.J., Adam, H., Mitchell, M.: Inclusivefacenet: Improving face attribute detection with race and gender diversity. arXiv preprint arXiv:1712.00193 (2017)
18. Yao, Y., Yu, H., Mu, J., Li, J., Pu, H.: Estimation of the gender ratio of chickens based on computer vision: Dataset and exploration. Entropy **22**(7), 719 (2020)

19. Zhang, Z., Song, Y., Qi, H.: Age progression/regression by conditional adversarial autoencoder. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5810–5818 (2017)