A. Appendix

A.1. Some notes on the code for reproducibility

In the code files that we provided, there are four important python files:

- $law\_data.py$: This file includes a function called $law\_data(seed)$ which processes the law school admission dataset and splits the dataset randomly into training and test datasets (we keep 70% of the datapoints for training). Later, in our experiments, we set the $seed$ equal to 0, 1, 2, 3, and 4 to get five different splits to repeat our experiments five times.
- $Adult\_data.py$: This file includes a function called $Adult\_dataset(seed)$ which processes the adult income dataset and splits the dataset randomly into training and test datasets. Later, in our experiments, we set the $seed$ equal to 0, 1, 2, 3, 4 to get five different splits to repeat our experiments five times.
- $Algorithms.py$: This file includes the following functions,
  - $ELminimizer(X0, Y0, X1, Y1, gamma, eta, model)$: This function implements `Elminimizer` algorithm. $(X0, Y0)$ are the training datapoints belonging to group $A = 0$ and $(X1, Y1)$ are the datapoits belonging to group $A = 1$. γ is the fairness level for EL constraint. η is the requlrarizer parameter (in experiment 1 and experiment 2, $\eta = 0.002$). $model$ determines the model that we want to train. If $model =$ "$linear$", then we are training a linear regression model. If $model =$ "$logistic$", then we train a logisitic regression model. This function returns five variables $(w, b, l0, l1, l)$. w, b are the weight vector and bias term of the trained model. l_0, l_1 are the average training loss of group 0 and group 1, respectively. l is the overall training loss.
  - $Algorithm2(X0, Y0, X1, Y1, gamma, eta, model)$: This function implements Algorithm 2 which calls `Elminimizer` algorithm twice. This function also returns five variables $(w, b, l0, l1, l)$. These variables have been defined above.
  - $Algorithm3(X0, Y0, X1, Y1, gamma, eta, model)$: This function implements Algorithm 3 which find a sub-optimal solution under EL fairness. This function also returns five variables $(w, b, l0, l1, l)$. These variables have been defined above.
  - $solve\_constrained\_opt(X0, Y0, X1, Y1, eta, landa, model)$: This function uses the CVXPY package to solve the optimization problem (4). We set $landa$ equal to $\lambda_{mid}^{(i)}$ to solve the optimization problem (4) in iteration i of Algorithm 1.
  - $calculate\_loss(w, b, X0, Y0, X1, Y1, model)$: This function is used to find the test loss. w, b are model parameters (trained by Algorithm 2 or 3). It returns the average loss of group 0 and group 1 and the overall loss based on the given dataset.
  - $solve\_lin\_constrained\_opt(X0, Y0, X1, Y1, gamma, eta, model)$: This function is for solving optimization (8) after linear relaxation.
- $Baseline.py$: this file include the following functions,
  - $penalty\_method(method, X\_0, y\_0, X\_1, y\_1, num\_itr, lr, r, gamma, seed, epsilon)$ where $method$ can be either "$linear$" for linear regression or "$logistic$" for logistic regression. This function uses the penalty method and trains the model under EL using the Adam optimization. num_itr is the maximum number of iterations. r is the regularization parameter (it is set to 0.002 in our experiment). lr is the learning rate and $gamma$ is the fairness level. ϵ is used for the stopping criterion. This function returns the trained model (which is a torch module), and training loss of group 0 and group 1, and the overall training loss. Please look at the next section for more details.
  - $fair\_batch(method, X\_0, y\_0, X\_1, y\_1, num\_itr, lr, r, alpha, gamma, seed, epsilon)$: This function is used to simulate the FairBAtch algorithm. The input parameters are similar to the input parameters of $penalty\_method$ except for $alpha$. This parameter determines how to adjust the sub-sampling distribution for mini-batch formation. Please look at the next section for more details.

    This function returns the trained model (which is a torch module), and training loss of group 0 and group 1, and the overall training loss.

We use the above functions to produce the results of experiment 1 and experiment 2. $table1\_2.py$ uses the above functions to reproduce the results in Table 1 and Table 2. $figure1\_2.py$ uses the above functions to reproduce figure 1 and figure 2. We

provide some comments in these files to make the code more readable. We have also provided code for training non-linear models. Please use $Table3.py$ and $figure3.py$ to generate the results in Table 3 and Figure 3, respectively.

Lastly, the following command to generate results in Table 1:

- `python3 table1_2.py --experiment=1 --gamma=0.0`
- `python3 table1_2.py --experiment=1 --gamma=0.1`

Use the following command to generate results in Table 2:

- `python3 table1_2.py --experiment=2 --gamma=0.0`
- `python3 table1_2.py --experiment=2 --gamma=0.1`

Use the following command to generate results in Table 3:

- `python3 table3.py --experiment=1 --gamma=0.0`
- `python3 table3.py --experiment=1 --gamma=0.1`

Use the following command to generate results in Figure 1:

- `python3 figure1_2.py --experiment=1`

Use the following command to generate results in Figure 2:

- `python3 figure1_2.py --experiment=2`

Use the following command to generate results in Figure 3:

- `python3 figure3.py --experiment=1`

Note that you need to install CVXPY, torch, and numpy.

A.2. Notes on FairBatch (Roh et al., 2020)

This method has been proposed to find a predictor under equal opportunity, equalized odd or statistical parity. In each epoch, this method identifies the disadvantaged group and increases the subsampling rate corresponding to the disadvantaged group in mini-batch selection for the next epoch. We modify this approach for γ-EL as follows,

- We initialize the sub-sampling rate of group a (denoted by $SR_a^{(0)}$) for mini-batch formation by $SR_a^{(0)} = \frac{n_a}{n}, a = 0, 1$. We Form the mini-batches using $SR_0^{(0)}$ and $SR_1^{(0)}$.
- At epoch i, we run gradient descent using the mini-batches formed by $SR_0^{(i-1)}$ and $SR_1^{(i-1)}$, and we obtain new model parameters $\boldsymbol{w}_i$.
- After epoch i, we calculate the empirical loss of each group. Then, we update $SR_a^{(i)}$ as follows,

$$\begin{aligned} SR_a^{(i)} &\longleftarrow SR_a^{(i-1)} + \alpha \qquad \text{if } \hat{L}_a(\boldsymbol{w}_i) - \hat{L}_{1-a}(\boldsymbol{w}_i) > \gamma \\ SR_a^{(i)} &\longleftarrow SR_a^{(i-1)} - \alpha \qquad \text{if } \hat{L}_a(\boldsymbol{w}_i) - \hat{L}_{1-a}(\boldsymbol{w}_i) < -\gamma \\ SR_a^{(i)} &\longleftarrow SR_a^{(i-1)} \qquad \text{o.w.,} \end{aligned}$$

  where is α is a hyperparameter and, in our experiment, is equal to 0.005 (used by (Roh et al., 2020)).

For the other hyperparameters in FairBatch, we also adopt those used in (Roh et al., 2020),

- Learning rate: 0.005
- Batch Size: 100
- Adam Optimization (Kingma and Ba, 2014) for gradient descent with default parameters in PyTorch

For stopping criteria, we stopped the learning process when the change in the objective function is less than 10^{-6} between two consecutive epochs.

A.3. Details of numerical experiments and additional numerical results

Due to the space limits of the main paper, we provide more details on our experiments here,

- Stopping criteria for penalty method and FairBatch: For stopping criteria, we stopped the learning process when the change in the objective function is less than 10^{-6} between two consecutive epochs. The reason that we used 10^{-6} was that we did not observe any significant change by choosing a smaller value.
- Learning rate for penalty method and FairBatch: We chose 0.005 for the learning rate. In the original implementation of FairBatch, the learning rate was 0.005 for linear models. Also, we realized that training a linear model is not sensitive to the learning rate.
- Stopping criteria for Algorithm 2 and Algorithm 3: As we stated in the main paper, we set $\epsilon = 0.01$ in `ELminimizer` and Algorithm 3. Choosing smaller ϵ did not change the performance significantly.
- Linear Relaxation: Note that equation (8) after linear relaxation is a convex optimization problem. We directly solve this optimization problem using CVXPY.

The experiment has been done on a system with the following configurations: 24 GB of RAM, 2 cores of P100-16GB GPU, and 2 cores of Intel Xeon CPU@2.3 GHz processor. We used GPUs for training FairBatch.

Experiments with a non-linear model To demonstrate how we can use our algorithms to fine-tune a non-linear model, we repeat the first experiment of Section 6 with nonlinear models. We work with Law School Admission dataset, and we train a Neural network with one hidden layer which consists of 125 neurons. We use sigmoid as the activation function for the hidden layer. We run the following algorithms for training,

- Penalty Method: We solve equation (10). Note that in this example, $\hat{L}$ and $\hat{L}_a$ are not convex anymore. The hyperparameters such as learning rate, penalty parameter, and stopping criterion, remain the same as in the first experiment.
- FairBatch: We train the whole network with mini-batch Adam optimization with a batch size of 100.
- Linear Relaxation: In order to take advantage of CVXPY, first, we train the network without any fairness constraint using Batch Adam optimization (batch size is equal to the size of the training dataset) and a learning rate of 0.005. Then, we fine-tune the parameters of the output layer. Note that the output layer has 126 parameters, and we fine tunes those under relaxed EL fairness. In particular, we solve optimization problem (9) after linear relaxation.
- Algorithm 2 and Algorithm 3: We can run Algorithm 2 to fine-tune the network. After training the network without any constraint, we solve (9) using Algorithm 2 and Algorithm 3.

Table 3 illustrates the means and standard deviations of empirical loss and the loss difference between Black and White students. The first row specifies desired fairness level ($\gamma = 0$ and $\gamma = 0.1$) used as the input to each algorithm. Both Algorithm 2 and Algorithm 3 can achieve a fairness level (i.e., $|\hat{L}_0 - \hat{L}_1|$) close to desired fairness level γ. Also, we can see that the MSE of Algorithm 2 and Algorithm 3 under the nonlinear model is slightly lower than the MSE under the linear model.

Table 3: Neural Network training under EL fairness. The loss function in this example is the mean squared error loss.

		$\gamma = 0$	$\gamma = 0.1$
PM	test loss	0.8358 ± 0.0154	0.8376 ± 0.0139
	test $\lvert\hat{L}_0 - \hat{L}_1\rvert$	0.6376 ± 0.0948	0.7708 ± 0.1207
LinRe	test loss	0.8430 ± 0.0150	0.8190 ± 0.0131
	test $\lvert\hat{L}_0 - \hat{L}_1\rvert$	0.6883 ± 0.0254	0.5872 ± 0.0344
Fair Batch	test loss	0.8674 ± 0.1032	0.8400 ± 0.0522
	test $\lvert\hat{L}_0 - \hat{L}_1\rvert$	0.1808 ± 0.0633	0.1995 ± 0.1053
ours Alg 2	test loss	0.9116 ± 0.0206	0.8552 ± 0.0163
	test $\lvert\hat{L}_0 - \hat{L}_1\rvert$	0.0806 ± 0.0547	0.1334 ± 0.0841
ours Alg 3	test loss	0.9468 ± 0.0198	0.8916 ± 0.0183
	test $\lvert\hat{L}_0 - \hat{L}_1\rvert$	0.0864 ± 0.0553	0.1422 ± 0.0893

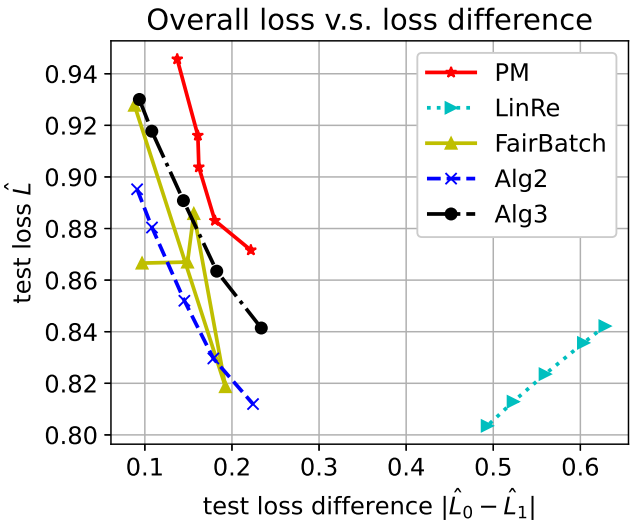

Figure 3: Trade-off between overall MSE and unfairness. A lower curve implies a better trade-off.

We also investigate how MSE $\hat{L}$ changes as a function of fairness level $|\hat{L}_1 - \hat{L}_0|$. Figure 3 illustrates the MSE-fairness trade-off. To generate this plot, we change choose $\gamma = [0.025, 0.05, 0.1, 0.15, 0.2]$. For each γ, we ran the experiment 5 times and calculated the average of MSE $\hat{L}$ and the average of MSE difference over 5 five runs using the test dataset. Based on Figure 3, we observe that FairBatch and LinRe are not very sensitive to the input γ. However, FairBatch may sometimes show a better trade-off than Algorithm 2. In this example, PM, Algorithm 2, and Algorithm 3 are very sensitive to γ, and as γ increases, MSE $\hat{L}$ decreases and $|\hat{L}_0 - \hat{L}_1|$ increases.

A.4. Notes on the Reduction Approach (Agarwal et al., 2018; 2019)

Let $Q(f)$ be a distribution over $\mathcal{F}$. In order to find optimal $Q(f)$ using the reduction approach, we have to solve the following optimization problem,

$$\begin{aligned}
\min_{Q} \quad & \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))\} \\
s.t., \quad & \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))|A=0\} = \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))\} \\
& \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))|A=1\} = \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))\}
\end{aligned}$$

Similar to (Agarwal et al., 2018; 2019), we can re-write the above optimization problem in the following form,

$$\begin{aligned}
\min_{Q} \quad & \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))\} \\
s.t., \quad & \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))|A=0\} - \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))\} \le 0 \\
& -\sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))|A=0\} + \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))\} \le 0 \\
& \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))|A=1\} - \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))\} \le 0 \\
& -\sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))|A=1\} + \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))\} \le 0
\end{aligned}$$

Then, the reduction approach forms the Lagrangian function, in the following form,

$$\begin{aligned}
L(Q,\mu) \quad = \quad & \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))\} \\
- \quad & \mu_1 \cdot \left(\sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))|A=0\} - \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))\} \right) \\
- \quad & \mu_2 \cdot \left(-\sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))|A=0\} + \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))\} \right) \\
- \quad & \mu_3 \cdot \left(\sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))|A=1\} - \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))\} \right) \\
- \quad & \mu_4 \cdot \left(-\sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))|A=1\} + \sum_{f\in\mathcal{F}} Q(f)\mathbb{E}\{l(Y,f(\boldsymbol{X}))\} \right), \\
& \mu_1 \ge 0, \mu_2 \ge 0, \mu_3 \ge 0, \mu_4 \ge 0.
\end{aligned}$$

Since f is parametrized with $\boldsymbol{w}$, we can find distribution $Q(\boldsymbol{w})$ over $\mathbb{R}^{d_{\boldsymbol{w}}}$. Therefore, we rewrite the problem in the following

form,

$$\begin{aligned} L(Q(\boldsymbol{w}), \mu_1, \mu_2, \mu_3, \mu_4) &= \sum_{\boldsymbol{w}} Q(\boldsymbol{w}) L(\boldsymbol{w}) \\ &- \mu_1 \left(\sum_{\boldsymbol{w}} Q(\boldsymbol{w}) L_0(\boldsymbol{w}) - \sum_{\boldsymbol{w}} Q(\boldsymbol{w}) L(\boldsymbol{w}) \right) \\ &- \mu_2 \left(- \sum_{\boldsymbol{w}} Q(\boldsymbol{w}) L_0(\boldsymbol{w}) + \sum_{\boldsymbol{w}} Q(\boldsymbol{w}) L(\boldsymbol{w}) \right) \\ &- \mu_3 \left(\sum_{\boldsymbol{w}} Q(\boldsymbol{w}) L_1(\boldsymbol{w}) - \sum_{\boldsymbol{w}} Q(\boldsymbol{w}) L(\boldsymbol{w}) \right) \\ &- \mu_4 \left(- \sum_{\boldsymbol{w}} Q(\boldsymbol{w}) L_1(\boldsymbol{w}) + \sum_{\boldsymbol{w}} Q(\boldsymbol{w}) L(\boldsymbol{w}) \right) \end{aligned}$$

The reduction approach updates $Q(\boldsymbol{w})$ and $(\mu_1, \mu_2, \mu_3, \mu_4)$ alternatively. Looking carefully at Algorithm 1 in (Agarwal et al., 2018), after updating $(\mu_1, \mu_2, \mu_3, \mu_4)$, we need to have access to an oracle that is able to solve the following optimization problem in each iteration,

$$\min_{\boldsymbol{w}} (1 + \mu_1 - \mu_2 + \mu_3 - \mu_4) L(\boldsymbol{w}) + (-\mu_1 + \mu_2) L_0(\boldsymbol{w}) + (-\mu_3 + \mu_4) L_1(\boldsymbol{w})$$

The above optimization problem is not convex for all $\mu_1, \mu_2, \mu_3, \mu_4$. Therefore, in order to use the reduction approach, we need to have access to an oracle that is able to solve the above non-convex optimization problem which is not available. Note that the original problem (1) is a non-convex optimization problem and using the reduction approach just leads to another non-convex optimization problem.

A.5. Equalized Loss & Bounded Group Loss

In this section, we study the relation between EL and BGL fairness notions. It is straightforward to see that any predictor satisfying γ-BGL also satisfies the γ-EL. However, it is unclear to what extend an *optimal* fair predictor under γ-EL satisfies the BGL fairness notion. Next, we theoretically study the relation between BGL and EL fairness notions.

Let $\boldsymbol{w}^*$ be denoted as the solution to (1) and $f_{\boldsymbol{w}^*}$ the corresponding optimal γ-EL fair predictor. Theorem A.1 below shows that under certain conditions, it is impossible for both groups to experience a loss larger than 2γ under the *optimal* γ-EL fair predictor.

Theorem A.1. *Suppose there exists a predictor that satisfies γ-BGL fairness notion. That is, the following optimization problem has at least one feasible point.*

$$\min_{\boldsymbol{w}} \; L(\boldsymbol{w}) \; s.t. \; L_a(\boldsymbol{w}) \leq \gamma, \; \forall a \in \{0, 1\}. \tag{12}$$

Then, the followings hold,

$$\begin{aligned} \min\{L_0(\boldsymbol{w}^*), L_1(\boldsymbol{w}^*)\} &\leq \gamma; \\ \max\{L_0(\boldsymbol{w}^*), L_1(\boldsymbol{w}^*)\} &\leq 2\gamma. \end{aligned}$$

Theorem A.1 shows that γ-EL implies 2γ-BGL if γ-BGL is a feasible constraint. Therefore, if γ is not too small (e.g., $\gamma = 0$), then EL and BGL are not contradicting each other.

We emphasize that we are not claiming that whether EL fairness is better than BGL. Instead, these relations indicate the impacts the two fairness constraints could have on the model performance; the results may further provide the guidance for policy-makers.

A.6. Proofs

In order to prove Theorem 3.3, we first introduce two lemmas.

Lemma A.2. *Under assumption 3.2, there exists* $\overline{\boldsymbol{w}} \in \mathbb{R}^{d_{\boldsymbol{w}}}$ *such that* $L_0(\overline{\boldsymbol{w}}) = L_1(\overline{\boldsymbol{w}}) = L(\overline{\boldsymbol{w}})$ *and* $\lambda^{(0)}_{start} \leq L(\overline{\boldsymbol{w}}) \leq \lambda^{(0)}_{end}$.

Proof. Let $q_0(\beta) = L_0((1-\beta)\boldsymbol{w}_{G_0} + \beta \boldsymbol{w}_{G_1})$ and $q_1(\beta) = L_1((1-\beta)\boldsymbol{w}_{G_0} + \beta \boldsymbol{w}_{G_1})$, and $q(\beta) = q_0(\beta) - q_1(\beta), \beta \in [0,1]$. Note that $\nabla_{\boldsymbol{w}} L_a(\boldsymbol{w}_{G_a}) = 0$ because $\boldsymbol{w}_{G_a}$ is the minimizer of $L_a(\boldsymbol{w})$.

First, we show that $L_0((1-\beta)\boldsymbol{w}_{G_0} + \beta \boldsymbol{w}_{G_1})$ is an increasing function in β, and $L_1((1-\beta)\boldsymbol{w}_{G_0} + \beta \boldsymbol{w}_{G_1})$ is a decreasing function in β. Note that $q_0'(0) = (\boldsymbol{w}_{G_1} - \boldsymbol{w}_{G_0})^T \nabla_{\boldsymbol{w}} L_0(\boldsymbol{w}_{G_0}) = 0$, and $q_0(\beta)$ is convex because $L_0(\boldsymbol{w})$ is convex. This implies that $q'(\beta)$ is an increasing function, and $q_0'(\beta) \geq 0, \forall \beta \in [0,1]$. Similarly, we can show that $q_1'(\beta) \leq 0, \forall \beta \in [0,1]$.

Note that under Assumption (3.2), $q(0) < 0$ and $q(1) > 0$. Therefore, by the intermediate value theorem, the exists $\overline{\beta} \in (0,1)$ such that $q(\overline{\beta}) = 0$. Define $\overline{\boldsymbol{w}} = (1-\overline{\beta})\boldsymbol{w}_{G_0} + \overline{\beta}\boldsymbol{w}_{G_1}$. We have,

$$
\begin{aligned}
q(\overline{\beta}) &= 0 \implies L_0(\overline{\boldsymbol{w}}) = L_1(\overline{\boldsymbol{w}}) = L(\overline{\boldsymbol{w}}) \\
\boldsymbol{w}_{G_0} &\text{ is } \text{minimizer of } L_0 \implies \\
L(\overline{\boldsymbol{w}}) &= L_0(\overline{\boldsymbol{w}}) \geq \lambda^{(0)}_{start} \\
q_0'(\beta) &\geq 0, \forall \beta \in [0,1] \implies q_0(1) \geq q_0(\overline{\beta}) \implies \\
\lambda^{(0)}_{end} &\geq L_0(\overline{\boldsymbol{w}}) = L(\overline{\boldsymbol{w}})
\end{aligned}
$$

Lemma A.3. $L_0(\boldsymbol{w}_i^*) = \lambda^{(i)}_{mid}$, *where* $\boldsymbol{w}_i^*$ *is the solution to* (4).

Proof. We proceed by contradiction. Assume that $L_0(\boldsymbol{w}_i^*) < \lambda^{(i)}_{mid}$ (i.e., $\boldsymbol{w}_i^*$ is an interior point of the feasible set of (4)). Notice that $\boldsymbol{w}_{G_1}$ cannot be in the feasible set of (4) because $L_0(\boldsymbol{w}_{G_1}) = \lambda^{(0)}_{end} > \lambda^{(i)}_{mid}$. As a result, $\nabla_{\boldsymbol{w}} L_1(\boldsymbol{w}_i^*) \neq 0$. This is a contradiction because $\boldsymbol{w}_i^*$ is an interior point of the feasible set of a convex optimization and cannot be optimal if $\nabla_{\boldsymbol{w}} L_1(\boldsymbol{w}_i^*)$ is not equal to zero.

Proof [Theorem 3.3]

Let $I_i = [\lambda^{(i)}_{start}, \lambda^{(i)}_{end}]$ be a sequence of intervals. It is easy to see that $I_1 \supseteq I_2 \supseteq \cdots$ and $\lambda^{(i)}_{end} - \lambda^{(i)}_{start} \to 0$ as $i \to \infty$. Therefore, by the Nested Interval Theorem, $\cap_{i=1}^{\infty} I_i$ consists of exactly one real number λ^*, and both $\lambda^{(i)}_{start}$ and $\lambda^{(i)}_{end}$ converge to λ^*. Because $\lambda^{(i)}_{mid} = \frac{\lambda^{(i)}_{start} + \lambda^{(i)}_{start}}{2}$, $\lambda^{(i)}_{mid}$ also converges to λ^*.

Now, we show that $L(\boldsymbol{w}^*) \in I_i$ for all i ($\boldsymbol{w}^*$ is the solution to (1) when $\gamma = 0$. As a result, $L_0(\boldsymbol{w}^*) = L_1(\boldsymbol{w}^*) = L(\boldsymbol{w}^*)$). Note that $L(\boldsymbol{w}^*) = L_0(\boldsymbol{w}^*) \geq \lambda^{(0)}_{start}$ because $\boldsymbol{w}_{G_0}$ is the minimizer of L_0. Moreover, $\lambda^{(0)}_{end} \geq L(\boldsymbol{w}^*)$ otherwise $L(\overline{\boldsymbol{w}}) < L(\boldsymbol{w}^*)$ ($\overline{\boldsymbol{w}}$ is defined in Lemma A.2) and $\boldsymbol{w}^*$ is not optimal solution under 0-EL. Therefore, $L(\boldsymbol{w}^*) \in I_0$.

Now we proceed by induction. Suppose $L(\boldsymbol{w}^*) \in I_i$. We show that $L(\boldsymbol{w}^*) \in I_{i+1}$ as well. We consider two cases.

- $L(\boldsymbol{w}^*) \leq \lambda^{(i)}_{mid}$. In this case $\boldsymbol{w}^*$ is a feasible point for (4), and $L_1(\boldsymbol{w}_i^*) = \lambda^{(i)} \leq L_1(\boldsymbol{w}^*) = L(\boldsymbol{w}^*) \leq \lambda^{(i)}_{mid}$. Therefore, $L(\boldsymbol{w}^*) \in I_{i+1}$.

- $L(\boldsymbol{w}^*) > \lambda^{(i)}_{mid}$. In this case, we proceed by contradiction to show that $\lambda^{(i)} \geq \lambda^{(i)}_{mid}$. Assume that $\lambda^{(i)} < \lambda^{(i)}_{mid}$. Define $r(\beta) = r_0(\beta) - r_1(\beta)$, where $r_a(\beta) = L_a((1-\beta)\boldsymbol{w}_{G_0} + \beta \boldsymbol{w}_i^*)$. Note that $\lambda^{(i)} = r_1(1)$ By Lemma A.3, $r_0(1) = \lambda^{(i)}_{mid}$. Therefore, $r(1) = \lambda^{(i)}_{mid} - \lambda^{(i)} > 0$. Moreover, under Assumption 3.2, $r(0) < 0$. Therefore, by the intermediate value theorem, there exists $\overline{\beta}_0 \in (0,1)$ such that $r(\overline{\beta}_0) = 0$. Similar to the proof of Lemma A.2, we can show that $r_0(\beta)$ in an increasing function for all $\beta \in [0,1]$. As a result $r_0(\overline{\beta}_0) < r_0(1) = \lambda^{(i)}_{mid}$. Define $\overline{\boldsymbol{w}}_0 = (1-\overline{\beta}_0)\boldsymbol{w}_{G_0} + \overline{\beta}_0 \boldsymbol{w}_i^*$. We have,

$$r_0(\overline{\beta}_0) = L_0(\overline{\boldsymbol{w}}_0) = L_1(\overline{\boldsymbol{w}}_0) = L(\overline{\boldsymbol{w}}_0) < \lambda^{(i)}_{mid} \tag{13}$$

$$L(\boldsymbol{w}^*) > \lambda^{(i)}_{mid} \tag{14}$$

  The last two equations imply that $\boldsymbol{w}^*$ is not a global optimal fair solution under 0-EL fairness constraint and it is not the global optmal solution to (1). This is a contradiction. Therefore, if $L(\boldsymbol{w}^*) > \lambda^{(i)}_{mid}$, then $\lambda^{(i)} \geq \lambda^{(i)}_{mid}$. As a result, $L(\boldsymbol{w}^*) \in I_{i+1}$

By two above cases and the nested interval theorem, we conclude that,

$$L(\boldsymbol{w}^*) \in \cap_{i=1}^{\infty} I_i, \ \lim_{i\to\infty} \lambda_{mid}^{(i)} = L(\boldsymbol{w}^*),$$

$$\text{define} \lambda_{mid}^{\infty} := \lim_{i\to\infty} \lambda_{mid}^{(i)}$$

Therefore, $\lim_{i\to\infty} \boldsymbol{w}_i^*$ would be the solution to the following optimziation problem,

$$\arg\min_{\boldsymbol{w}} L_1(\boldsymbol{w}) s.t., L_0(\boldsymbol{w}) \leq \lambda_{mid}^{\infty} = L(\boldsymbol{w}^*)$$

By lemma A.3, the solution to above optimization problem (i.e., $\lim_{i\to\infty} \boldsymbol{w}_i^*$) satisfies the following, $L_0(\lim_{i\to\infty} \boldsymbol{w}_i^*) = \lambda_{mid}^{\infty} = L(\boldsymbol{w}^*)$. Therefore, $\lim_{i\to\infty} \boldsymbol{w}_i^*$ is the global optimal solution to optimization problem (1).

Proof [Theorem 3.4] Let's assume that $\boldsymbol{w}_O$ does not satisfy the γ-EL.[11] Let $\boldsymbol{w}^*$ be the optimal weight vector under γ-EL. It is clear that $\boldsymbol{w}^* \neq \boldsymbol{w}_O$.

Step 1. we show that one of the following holds,

$$L_0(\boldsymbol{w}^*) - L_1(\boldsymbol{w}^*) = \gamma \tag{15}$$

$$L_0(\boldsymbol{w}^*) - L_1(\boldsymbol{w}^*) = -\gamma \tag{16}$$

Proof by contradiction. Assume $-\gamma < L_0(\boldsymbol{w}^*) - L_1(\boldsymbol{w}^*) < \gamma$. This implies that $\boldsymbol{w}^*$ is an interior point of the feasible set of optimization problem (1). Since $\boldsymbol{w}^* \neq \boldsymbol{w}_O$, then $\nabla L(\boldsymbol{w}^*) \neq 0$. As a result, object function of (1) can be improved at $\boldsymbol{w}^*$ by moving toward $-\nabla L(\boldsymbol{w}^*)$. This a contradiction. Therefore, $|L_0(\boldsymbol{w}^*) - L_1(\boldsymbol{w}^*)| = \gamma$.

Step 2. Function $\boldsymbol{w}_\gamma = \texttt{ELminimizer}(\boldsymbol{w}_{G_0}, \boldsymbol{w}_{G_0}, \epsilon, \gamma)$ is the solution to the following optimization problem,

$$\min_{\boldsymbol{w}} \Pr\{A=0\} L_0(\boldsymbol{w}) + \Pr\{A=1\} L_1(\boldsymbol{w}),$$
$$s.t., L_0(\boldsymbol{w}) - L_1(\boldsymbol{w}) = \gamma \tag{17}$$

To show the above claim, notice that the solution to optimization problem (17) is the same as the following,

$$\min_{\boldsymbol{w}} \Pr\{A=0\} L_0(\boldsymbol{w}) + \Pr\{A=1\} \tilde{L}_1(\boldsymbol{w}),$$
$$s.t., L_0(\boldsymbol{w}) - \tilde{L}_1(\boldsymbol{w}) = 0, \tag{18}$$

where $\tilde{L}_1(\boldsymbol{w}) = L_1(\boldsymbol{w}) + \gamma$. Since $L_0(\boldsymbol{w}_{G_0}) - \tilde{L}_1(\boldsymbol{w}_{G_0}) < 0$ and $L_0(\boldsymbol{w}_{G_1}) - \tilde{L}_1(\boldsymbol{w}_{G_1}) > 0$, by Theorem 3.3, we know that $\boldsymbol{w}_\gamma = \texttt{ELminimizer}(\boldsymbol{w}_{G_0}, \boldsymbol{w}_{G_0}, \epsilon, \gamma)$ find the solution to (18) when ϵ goes to zero.

Lastly, because $|L_0(\boldsymbol{w}^*) - L_1(\boldsymbol{w}^*)| = \gamma$, we have,

$$\boldsymbol{w}^* = \begin{cases} \boldsymbol{w}_\gamma & \text{if } L(\boldsymbol{w}_\gamma) \leq L(\boldsymbol{w}_{-\gamma}) \\ \boldsymbol{w}_{-\gamma} & \text{o.w.} \end{cases} \tag{19}$$

Thus, Algorithm 2 finds the solution to (1).

Proof [Theorem 4.1]

1. Under Assumption 3.2, $g(1) < 0$. Moreover, $g(0) \geq 0$. Therefore, by the intermediate value theorem, there exists $\beta_0 \in [0,1]$ such that $g(\beta_0) = 0$.
2. Since $\boldsymbol{w}_O$ is the minimizer of $L(\boldsymbol{w})$, $h'(0) = 0$. Moreover, since $L(\boldsymbol{w})$ is strictly convex, $h(\beta)$ is strictly convex and $h'(\beta)$ is strictly increasing function. As a result, $h'(\beta) > 0$ for $\beta > 0$, and $h(\beta)$ is strictly increasing.
3. Similar to the above argument, $s(\beta) = L_{\hat{a}}((1-\beta)\boldsymbol{w}_O + \beta \boldsymbol{w}_{G_{\hat{a}}})$ is strictly decreasing function (notice that $s'(1) = 0$ and $s(\beta)$ is strictly convex).

[11] If $\boldsymbol{w}_O$ satisfies γ-EL, it will be the optimal predictor under γ-EL fairness. Therefore, there is no need to solve any constrained optimization problem. Note that $\boldsymbol{w}_O$ is the solution to problem (7).

Note that since $h(\beta) = \Pr\{A = \hat{a}\}L_{\hat{a}}((1-\beta)\boldsymbol{w}_O + \beta\boldsymbol{w}_{G_{\hat{a}}}) + \Pr\{A = 1-\hat{a}\}L_{1-\hat{a}}((1-\beta)\boldsymbol{w}_O + \beta\boldsymbol{w}_{G_{\hat{a}}})$ is strictly increasing and $L_{\hat{a}}((1-\beta)\boldsymbol{w}_O + \beta\boldsymbol{w}_{G_{\hat{a}}})$ is strictly decreasing. Therefore, we conclude that $L_{1-\hat{a}}((1-\beta)\boldsymbol{w}_O + \beta\boldsymbol{w}_{G_{\hat{a}}})$ is strictly increasing. As a result, $g(\beta)$ should be strictly decreasing.

Proof [Theorem 4.2] First, we show that if $g_\gamma(0) \leq 0$, then $\boldsymbol{w}_O$ satisfies γ-EL.

$$g_\gamma(0) \leq 0 \implies g(\beta) - \gamma \leq 0 \implies L_{\hat{a}}(\boldsymbol{w}_O) - L_{1-\hat{a}}(\boldsymbol{w}_O) \leq \gamma$$

Moreover, $L_{\hat{a}}(\boldsymbol{w}_O) - L_{1-\hat{a}}(\boldsymbol{w}_O) \geq 0$ because $\hat{a} = \arg\max_a L_a(\boldsymbol{w}_O)$. Therefore, γ-EL is satisfied.

Now, let $g_\gamma(0) > 0$. Note that under Assumption 3.2, $g_\gamma(1) = L_{\hat{a}}(\boldsymbol{w}_{G_{\hat{a}}}) - L_{1-\hat{a}}(\boldsymbol{w}_{G_{\hat{a}}}) - \gamma < 0$. Therefore, by the intermediate value theorem, there exists β_0 such that $g_\gamma(\beta_0) = 0$. Moreover, based on Theorem 4.2, g_γ is a strictly decreasing function. Therefore, the binary search proposed in Algorithm 3 converges to the root of $g_\gamma(\beta)$. As a result, $(1-\beta^{(\infty)}_{mid})\boldsymbol{w}_O + \beta^{(\infty)}_{mid}\boldsymbol{w}_{G_{\hat{a}}}$ satisfies γ-EL. Note that since $g(\beta)$ is strictly decreasing, and $g(\beta^{(\infty)}_{mid}) = \gamma$, and $\beta^{(\infty)}_{mid}$ is the smallest possible β under which $(1-\beta)\boldsymbol{w}_O + \beta\boldsymbol{w}_{G_{\hat{a}}}$ satisfies γ-EL. Since h is increasing, the smallest possible β gives us a better accuracy.

Proof [Theorem 4.3] If $g_\gamma(0) \leq 0$, then $\boldsymbol{w}_O$ satisfies γ-EL, and $\underline{\boldsymbol{w}} = \boldsymbol{w}_O$. In this case, it is easy to see that $L(\boldsymbol{w}_O) \leq \max_{a\in\{0,1\}} L_a(\boldsymbol{w}_O)$ (because $L(\boldsymbol{w}_O)$ is a weighted average of $L_0(\boldsymbol{w}_O)$ and $L_1(\boldsymbol{w}_O)$).

Now assume that $g_\gamma(0) > 0$. Note that if we prove this theorem for $\gamma = 0$, then the theorem will hold for $\gamma > 0$. This is because the **optimal** predictor under 0-EL satisfies γ-EL condition as well. In other words, 0-EL is a stronger constraint than γ-EL.

Let $\gamma = 0$. In this case, Algorithm 3 finds $\underline{\boldsymbol{w}} = (1-\beta_0)\boldsymbol{w}_O + \beta_0\boldsymbol{w}_{G_{\hat{a}}}$, where β_0 is defined in Theorem 4.1. We have,

$$(*)\ \ g(\beta_0) = 0 = L_{\hat{a}}(\underline{\boldsymbol{w}}) - L_{1-\hat{a}}(\underline{\boldsymbol{w}})$$

In the proof of theorem 4.1, we showed that $L_{\hat{a}}((1-\beta)\boldsymbol{w}_O + \beta\boldsymbol{w}_{G_{\hat{a}}})$ is decreasing in β. Therefore, we have,

$$(**)\ \ L_{\hat{a}}(\underline{\boldsymbol{w}}) \leq L_{\hat{a}}(\boldsymbol{w}_O)$$

Therefore, we have,

$$\begin{aligned} L(\underline{\boldsymbol{w}}) &= \Pr(A=0)\cdot L_{\hat{a}}(\underline{\boldsymbol{w}}) + (1-\Pr(A=1))\cdot L_{1-\hat{a}}(\underline{\boldsymbol{w}}) && (20)\\ (\text{By } (*)) \quad &= L_{\hat{a}}(\underline{\boldsymbol{w}}) && (21)\\ (\text{By } (**)) \quad &\leq L_{\hat{a}}(\boldsymbol{w}_O) && (22)\end{aligned}$$

Proof [Theorem 4.5]

By the triangle inequality, the following holds,

$$\sup_{f_{\boldsymbol{w}}\in\mathcal{F}} ||L_0(\boldsymbol{w}) - L_1(\boldsymbol{w})| - |\hat{L}_0(\boldsymbol{w}) - \hat{L}_1(\boldsymbol{w})|| \leq \tag{23}$$

$$\sup_{f_{\boldsymbol{w}}\in\mathcal{F}} |L_0(\boldsymbol{w}) - \hat{L}_0(\boldsymbol{w})| + \sup_{f_{\boldsymbol{w}}\in\mathcal{F}} |L_1(\boldsymbol{w}) - \hat{L}_1(\boldsymbol{w})|. \tag{24}$$

Therefore, with probability at least $1-2\delta$ we have,

$$\sup_{f_{\boldsymbol{w}}\in\mathcal{F}} ||L_0(\boldsymbol{w}) - L_1(\boldsymbol{w})| - |\hat{L}_0(\boldsymbol{w}) - \hat{L}_1(\boldsymbol{w})|| \leq B(\delta, n_0, \mathcal{F}) + B(\delta, n_1, \mathcal{F}) \tag{25}$$

As a result, with probability $1-2\delta$ holds,

$$\{\boldsymbol{w} | f_{\boldsymbol{w}} \in \mathcal{F}, |L_0(\boldsymbol{w}) - L_1(\boldsymbol{w})| \leq \gamma\} \subseteq \{\boldsymbol{w} | f_{\boldsymbol{w}} \in \mathcal{F}, |\hat{L}_0(\boldsymbol{w}) - \hat{L}_1(\boldsymbol{w})| \leq \hat{\gamma}\} \tag{26}$$

Now consider the following,

$$L(\hat{\boldsymbol{w}}) - L(\boldsymbol{w}^*) = L(\hat{\boldsymbol{w}}) - \hat{L}(\hat{\boldsymbol{w}}) + \hat{L}(\hat{\boldsymbol{w}}) - \hat{L}(\boldsymbol{w}^*) + \hat{L}(\boldsymbol{w}^*) - L(\boldsymbol{w}^*) \tag{27}$$

By (26), $\hat{L}(\hat{\boldsymbol{w}}) - \hat{L}(\boldsymbol{w}^*) \leq 0$ with probability $1 - 2\delta$. Thus, with probability at least $1 - 2\delta$, we have,

$$L(\hat{\boldsymbol{w}}) - L(\boldsymbol{w}^*) \leq L(\hat{\boldsymbol{w}}) - \hat{L}(\hat{\boldsymbol{w}}) + \hat{L}(\boldsymbol{w}^*) - L(\boldsymbol{w}^*). \tag{28}$$

Therefore, under assumption 4.4, we can conclude with probability at least $1 - 6\delta$, $L(\hat{\boldsymbol{w}}) - L(\boldsymbol{w}^*) \leq 2B(\delta, n, \mathcal{F})$. In addition, by (25), with probability at least $1 - 2\delta$, we have,

$$\begin{aligned} |L_0(\hat{\boldsymbol{w}}) - L_1(\hat{\boldsymbol{w}})| &\leq B(\delta, n_0, \mathcal{F}) + B(\delta, n_1, \mathcal{F}) + |\hat{L}_0(\boldsymbol{w}) - \hat{L}_1(\boldsymbol{w})| \\ &\leq \hat{\gamma} + B(\delta, n_0, \mathcal{F}) + B(\delta, n_1, \mathcal{F}) \\ &= \gamma + 2B(\delta, n_0, \mathcal{F}) + 2B(\delta, n_1, \mathcal{F}) \end{aligned}$$

Proof [Theorem A.1] Let $\tilde{\boldsymbol{w}}$ be a feasible point of optimization problem (12), then $\tilde{\boldsymbol{w}}$ is also a feasible point to (1).

We proceed by contradiction. We consider three cases,

- If $\min\{L_0(\boldsymbol{w}^*), L_1(\boldsymbol{w}^*)\} > \gamma$ and $\max\{L_0(\boldsymbol{w}^*), L_1(\boldsymbol{w}^*)\} > 2\gamma$. In this case,
$$L(\boldsymbol{w}^*) > \gamma \geq L(\tilde{\boldsymbol{w}}).$$
This is a contradiction because it implies that $\boldsymbol{w}^*$ is not an optimal solution to (1), and $\tilde{\boldsymbol{w}}$ is a better solution for (1).
- If $\min\{L_0(\boldsymbol{w}^*), L_1(\boldsymbol{w}^*)\} > \gamma$ and $\max\{L_0(\boldsymbol{w}^*), L_1(\boldsymbol{w}^*)\} \leq 2\gamma$. This case is similar to above. $\min\{L_0(\boldsymbol{w}^*), L_1(\boldsymbol{w}^*)\} > \gamma$ implies that $L(\boldsymbol{w}^*) > \gamma \geq L(\tilde{\boldsymbol{w}})$. This is a contradiction because it implies that $\boldsymbol{w}^*$ is not an optimal solution to (1).
- If $\min\{L_0(\boldsymbol{w}^*), L_1(\boldsymbol{w}^*)\} \leq \gamma$ and $\max\{L_0(\boldsymbol{w}^*), L_1(\boldsymbol{w}^*)\} > 2 \cdot \gamma$. We have:
$$\max\{L_0(\boldsymbol{w}^*), L_1(\boldsymbol{w}^*)\} - \min\{L_0(\boldsymbol{w}^*), L_1(\boldsymbol{w}^*)\} > \gamma,$$
which shows that $\boldsymbol{w}^*$ is not a feasible point for (1). This is a contradiction.

Therefore, $\max\{L_0(\boldsymbol{w}^*), L_1(\boldsymbol{w}^*)\} \leq 2\gamma$ and $\min\{L_0(\boldsymbol{w}^*), L_1(\boldsymbol{w}^*)\} \leq \gamma$.