Appendix A Proofs

A.1 Proof of Theorem 1

Proof of Theorem 1. The metric in Eq. (3) can be written in a variable notation as:

$$\text{metric}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_j \frac{a_j \sum_i \hat{y}_i y_i + b_j \sum_i (1-\hat{y}_i)(1-y_i) + f_j(\sum_i \hat{y}_i, \sum_i y_i)}{g_j(\sum_i \hat{y}_i, \sum_i y_i)}. \tag{17}$$

Therefore, the expected value of the metrics can be computed as:

$$\mathbb{E}_{\mathcal{P}(\hat{\mathbf{Y}});\mathcal{Q}(\check{\mathbf{Y}})} \Big[\text{metric}(\hat{\mathbf{Y}}, \check{\mathbf{Y}})\Big] \tag{18}$$

$$\stackrel{a}{=} \sum_{\hat{\mathbf{y}} \in \{0,1\}^n} \sum_{\check{\mathbf{y}} \in \{0,1\}^n} \mathcal{P}(\hat{\mathbf{y}}) \mathcal{Q}(\check{\mathbf{y}}) \, \text{metric}(\hat{\mathbf{y}}, \check{\mathbf{y}}) \tag{19}$$

$$\stackrel{b}{=} \sum_{\hat{\mathbf{y}} \in \{0,1\}^n} \sum_{\check{\mathbf{y}} \in \{0,1\}^n} \mathcal{P}(\hat{\mathbf{y}}) \mathcal{Q}(\check{\mathbf{y}}) \sum_j \frac{a_j \sum_i \hat{y}_i \check{y}_i + b_j \sum_i (1-\hat{y}_i)(1-\check{y}_i) + f_j(\sum_i \hat{y}_i, \sum_i \check{y}_i)}{g_j(\sum_i \hat{y}_i, \ \sum_i \check{y}_i)} \tag{20}$$

$$\stackrel{c}{=} \sum_{k \in [0,n]} \sum_{l \in [0,n]} \sum_{\{\hat{\mathbf{y}} | \Sigma_i \hat{y}_i = k\}} \sum_{\{\check{\mathbf{y}} | \Sigma_i \check{y}_i = l\}} \mathcal{P}(\hat{\mathbf{y}}) \mathcal{Q}(\check{\mathbf{y}}) \Big(\sum_j \frac{a_j \sum_i \hat{y}_i \check{y}_i + b_j \sum_i (1-\hat{y}_i)(1-\check{y}_i) + f_j(k,l)}{g_j(k, \ l)} \Big) \tag{21}$$

$$\stackrel{d}{=} \sum_{k \in [0,n]} \sum_{l \in [0,n]} \sum_j \frac{1}{g_j(k,l)} \Big(a_j \sum_{\{\hat{\mathbf{y}} | \Sigma_i \hat{y}_i = k\}} \sum_{\{\check{\mathbf{y}} | \Sigma_i \check{y}_i = l\}} \mathcal{P}(\hat{\mathbf{y}}) \mathcal{Q}(\check{\mathbf{y}}) \sum_i \hat{y}_i \check{y}_i \tag{22}$$

$$+ b_j \sum_{\{\hat{\mathbf{y}} | \Sigma_i \hat{y}_i = k\}} \sum_{\{\check{\mathbf{y}} | \Sigma_i \check{y}_i = l\}} \mathcal{P}(\hat{\mathbf{y}}) \mathcal{Q}(\check{\mathbf{y}}) \sum_i (1-\hat{y}_i)(1-\check{y}_i)$$

$$+ \sum_{\{\hat{\mathbf{y}} | \Sigma_i \hat{y}_i = k\}} \sum_{\{\check{\mathbf{y}} | \Sigma_i \check{y}_i = l\}} \mathcal{P}(\hat{\mathbf{y}}) \mathcal{Q}(\check{\mathbf{y}}) f_j(k,l) \Big)$$

$$\stackrel{e}{=} \sum_{k \in [0,n]} \sum_{l \in [0,n]} \sum_j \frac{1}{g_j(k,l)} \Big(a_j \sum_i \mathcal{P}(\hat{y}_i = 1, \textstyle\sum_{i'} \hat{y}_{i'} = k) \mathcal{Q}(\check{y}_i = 1, \sum_{i'} \check{y}_{i'} = l) \tag{23}$$

$$+ b_j \sum_i (\mathcal{P}(\hat{y}_i = 0, \textstyle\sum_{i'} \hat{y}_{i'} = k)) (\mathcal{Q}(\check{y}_i = 0, \sum_{i'} \check{y}_{i'} = l)) + f_j(k,l) \mathcal{P}(\textstyle\sum_i \hat{y}_i = k) \mathcal{Q}(\sum_i \check{y}_i = l) \Big)$$

$$\stackrel{f}{=} \sum_{k \in [0,n]} \sum_{l \in [0,n]} \sum_j \frac{1}{g_j(k,l)} \big(a_j [\mathbf{p}_k^1 \cdot \mathbf{q}_l^1] + b_j [\mathbf{p}_k^0 \cdot \mathbf{q}_l^0] + f_j(k,l) r_k s_l \big). \tag{24}$$

The transformations above are explained as follow:

(a) Expanding the definition of expectation of the metric to the sum of probability-weighted metrics.

(b) Applying the construction of our performance metric.

(c) Grouping the values of the metric in terms of $\sum_i \hat{y}_i = k$ and $\sum_i \check{y}_i = l$.

(d) Since each f_j is just a linear function over $\sum_i \hat{y}_i \check{y}_i$ and $\sum_i (1-\hat{y}_i)(1-\check{y}_i)$, we can push the summation over $\sum_{\{\hat{\mathbf{y}} | \Sigma_i \hat{y}_i = k\}} \sum_{\{\check{\mathbf{y}} | \Sigma_i \check{y}_i = l\}}$ inside f_j.

(e) Since $\sum_i \hat{y}_i \check{y}_i$ and $\sum_i (1-\hat{y}_i)(1-\check{y}_i)$ are both decomposable, then the expectation over $\mathcal{P}(\hat{\mathbf{y}})$ and $\mathcal{Q}(\check{\mathbf{y}})$ for the case where $\sum_i \hat{y}_i = k$ and $\sum_i \check{y}_i = l$ can be decomposed into each individual marginal probabilities $\mathcal{P}(\hat{y}_i, \sum_{i'} \hat{y}_{i'} = k)$ and $\mathcal{Q}(\check{y}_i, \sum_{i'} \check{y}_{i'} = l)$. Similarly, given fixed k and l, $f_j(k,l)$ is just a constant. Hence we can simplify the expectation over $f_j(k,l)$ in terms of the marginal probabilities of $\mathcal{P}(\sum_i \hat{y}_i = k)$ and $\mathcal{Q}(\sum_i \check{y}_i = l)$.

(f) Rewriting the marginal probabilities in vector notations.

□

A.2 Proof of Theorem 2

Proof of Theorem 2. From Theorem 1 we know that:

$$\max_{\theta} \mathbb{E}_{\tilde{P}(\mathbf{X},\mathbf{Y})}\left[\min_{\mathcal{Q}(\check{\mathbf{Y}})}\max_{\mathcal{P}(\hat{\mathbf{Y}})} \mathbb{E}_{\mathcal{P}(\hat{\mathbf{Y}});\mathcal{Q}(\check{\mathbf{Y}})}\left[\text{metric}(\hat{\mathbf{Y}},\check{\mathbf{Y}}) - \theta^{\intercal}\left(\phi(\mathbf{X},\check{\mathbf{Y}}) - \phi(\mathbf{X},\mathbf{Y})\right)\right]\right] \tag{25}$$

$$= \max_{\theta} \mathbb{E}_{\tilde{P}(\mathbf{X},\mathbf{Y})}\left[\min_{\mathcal{Q}(\check{\mathbf{Y}})}\max_{\mathcal{P}(\hat{\mathbf{Y}})}\left[\sum_{k\in[0,n]}\sum_{l\in[0,n]}\sum_{j}\tfrac{1}{g_j(k,l)}\left\{a_j[\mathbf{p}_k^1\cdot\mathbf{q}_l^1] + b_j[\mathbf{p}_k^0\cdot\mathbf{q}_l^0] + f_j(k,l)r_k s_l\right\}\right.\right. \tag{26}$$

$$\left.\left. - \mathbb{E}_{\mathcal{Q}(\check{\mathbf{Y}})}\left[\theta^{\intercal}\left(\phi(\mathbf{X},\check{\mathbf{Y}}) - \phi(\mathbf{X},\mathbf{Y})\right)\right]\right].$$

Note that the values for some $\mathbf{p}_k^a$ and $\mathbf{q}_l^a$ are known, i.e.:

$$(\mathbf{p}_0^1)_i = \mathcal{P}(\hat{y}_i = 1, \textstyle\sum_i \hat{y}_i = 0)) = 0, \quad \forall i \in [1,n] \tag{27}$$

$$(\mathbf{p}_n^0)_i = \mathcal{P}(\hat{y}_i = 0, \textstyle\sum_i \hat{y}_i = n)) = 0, \quad \forall i \in [1,n] \tag{28}$$

$$(\mathbf{p}_n^1)_i = \mathcal{P}(\hat{y}_i = 1, \textstyle\sum_i \hat{y}_i = n)) = \mathcal{P}(\mathbf{1}), \quad \forall i \in [1,n] \tag{29}$$

$$(\mathbf{p}_0^0)_i = \mathcal{P}(\hat{y}_i = 0, \textstyle\sum_i \hat{y}_i = 0)) = \mathcal{P}(\mathbf{0}), \quad \forall i \in [1,n] \tag{30}$$

and similarly for $\mathbf{q}_l^a$.

We now analyze the relation between $\mathbf{p}_k^1$ and $\mathbf{p}_k^0$ (which also applies to $\mathbf{q}_k^1$ and $\mathbf{q}_k^0$). Note that each $\mathcal{P}(\hat{\mathbf{y}})$ such that $\sum_i \hat{y}_i = k$ appears k times in $\sum_i \mathcal{P}(\hat{y}_i = 1, \sum_i \hat{y}_i = k))$, which implies:

$$r_k = \mathcal{P}(\textstyle\sum_i \hat{y}_i = k) = \frac{1}{k}\sum_i \mathcal{P}(\hat{y}_i = 1, \sum_i \hat{y}_i = k)). \tag{31}$$

Therefore, we also have the relation:

$$\begin{aligned}\mathcal{P}(\hat{y}_i = 0, \textstyle\sum_i \hat{y}_i = k) =& \mathcal{P}(\textstyle\sum_i \hat{y}_i = k) - \mathcal{P}(\hat{y}_i = 1, \textstyle\sum_i \hat{y}_i = k) \\ =& \tfrac{1}{k}\textstyle\sum_i \mathcal{P}(\hat{y}_i = 1, \sum_i \hat{y}_i = k)) - \mathcal{P}(\hat{y}_i = 1, \textstyle\sum_i \hat{y}_i = k),\end{aligned}$$

for all $k \in [1, n-1]$. In vector notation, we can write:

$$r_k = \tfrac{1}{k}(\mathbf{p}_k^1 \cdot \mathbf{1}) \tag{32}$$

$$\mathbf{p}_k^0 = \tfrac{1}{k}(\mathbf{p}_k^1 \cdot \mathbf{1})\mathbf{1} - \mathbf{p}_k^1, \qquad \forall k \in [1, n-1]. \tag{33}$$

We know already that $\mathbf{p}_n^0 = \mathbf{0}$. For computing $\mathbf{p}_0^0$, we know that $\mathcal{P}(\hat{y}_i = 0, \sum_i \hat{y}_i = 0) = \mathcal{P}(\sum_i \hat{y}_i = 0) = \mathcal{P}(\mathbf{0})$ which can be computed as:

$$\begin{aligned}\mathcal{P}(\mathbf{0}) &= 1 - \textstyle\sum_{k\in[1,n]} \mathcal{P}(\sum_i \hat{y}_i = k) \\ &= 1 - \textstyle\sum_{k\in[1,n]} \frac{1}{k}\sum_i \mathcal{P}(\hat{y}_i = 1, \sum_i \hat{y}_i = k)) \\ &= 1 - \textstyle\sum_{k\in[1,n]} \frac{\mathbf{p}_k^1 \cdot \mathbf{1}}{k}\end{aligned} \tag{34}$$

Therefore, we can compute all values in $\mathbf{p}_k^0, \forall k \in [0,n]$, r_k, $\mathcal{P}(\mathbf{0})$, and $\mathcal{P}(\mathbf{1})$ from $\mathbf{p}_k^1$, and thus we can perform optimization over $\mathbf{p}_k^1$ and $\mathbf{q}_k^1$ only. For short, we write the as just $\mathbf{p}_k$ and $\mathbf{q}_k$. Note that we know that $\mathbf{p}_0 = \mathbf{q}_0 = \mathbf{0}$. Therefore, it suffices to optimize only over $\mathbf{p}_k$ and $\mathbf{q}_k$, for all $k \in [1,n]$. Let us denote a $n \times n$ matrix $\mathbf{P}$ where each column $\mathbf{P}_{(:,k)}$ represents $\mathbf{p}_k$. Similarly, we denote a matrix $\mathbf{Q}$ for $\mathbf{q}_k$.

Let us take a look at the property of the marginal probability matrices $\mathbf{P}$ and $\mathbf{Q}$. To be a valid marginal probability, $\mathbf{P}$ has to satisfy the following constraints:

$$p_{i,k} \geq 0 \quad \forall i,k \in [1,n] \tag{35}$$

$$\textstyle\sum_k p_{i,k} \leq 1 \quad \forall i \in [1,n] \tag{36}$$

$$p_{i,k} \leq \tfrac{1}{k}\textstyle\sum_j p_{j,k} \quad \forall i,k \in [1,n] \tag{37}$$

$$\textstyle\sum_k \frac{1}{k}\sum_i p_{i,k} \leq 1 \tag{38}$$

The constraints above are described below:

- The first constraint is for the non-negativity of probability.
- The second constraint is from $\mathcal{P}(\hat{y}_i = 1) = \sum_k \mathcal{P}(\hat{y}_i = 1, \sum_i \hat{y}_i = k) \leq 1$.
- The third constraint comes from the fact that each $\mathcal{P}(\hat{\mathbf{y}})$ such that $\sum_i \hat{y}_i = k$ appears k times in $\sum_i \mathcal{P}(\hat{y}_i = 1, \sum_i \hat{y}_i = k))$, and thus, $\mathcal{P}(\sum_i \hat{y}_i = k) = \frac{1}{k} \sum_i \mathcal{P}(\hat{y}_i = 1, \sum_i \hat{y}_i = k))$. Therefore, the inequality of $\mathcal{P}(y_i = 1, \sum_i \hat{y}_i = k) \leq \mathcal{P}(\sum_i \hat{y}_i = k)$ must hold which implies the third constraint.
- The fourth constraint comes from the fact that $\sum_k \mathcal{P}(\sum_i \hat{y}_i = k) \leq 1$.

The same constraints also need to hold for the probability matrix $\mathbf{Q}$. We can also see that satisfying the third and fourth constraints implies the second constraints, i.e.:

$$\sum_k p_{i,k} \leq \sum_k \tfrac{1}{k} \textstyle\sum_j p_{j,k} \leq 1. \tag{39}$$

Now we take a look at the features. Let the pair $(\mathbf{x}, \mathbf{y})$ be the empirical training data. Based on the construction of our features, we compute the potentials for $\theta^\intercal \phi(\mathbf{x}, \mathbf{y})$ as:

$$\theta^\intercal \phi(\mathbf{x}, \mathbf{y}) = \theta^\intercal \sum_i \phi(\mathbf{x}, y_i) = \theta^\intercal \sum_i \mathbb{I}[y_i = 1] \phi(\mathbf{x}, y_i = 1) = \langle \mathbf{y}, \Psi^\intercal \theta \rangle, \tag{40}$$

and the potentials for $\mathbb{E}_{\mathcal{Q}(\check{\mathbf{Y}})} \left[\theta^\intercal \phi(\mathbf{x}, \check{\mathbf{Y}})\right]$ as:

$$\mathbb{E}_{\mathcal{Q}(\check{\mathbf{Y}})} \left[\theta^\intercal \phi(\mathbf{x}, \check{\mathbf{Y}})\right] = \mathbb{E}_{\mathcal{Q}(\check{\mathbf{Y}})} \left[\theta^\intercal \sum_i \phi(\mathbf{x}, \check{Y}_i)\right] = \theta^\intercal \sum_i \mathcal{Q}(\check{y}_i = 1) \phi(\mathbf{x}, \check{y}_i = 1) = \langle \mathbf{Q}^\intercal \mathbf{1}, \Psi^\intercal \theta \rangle. \tag{41}$$

Therefore, we can simplify Eq. (2) as:

$$\max_\theta \left\{ \min_{\mathbf{Q} \in \Delta} \max_{\mathbf{P} \in \Delta} \left[\sum_{k \in [0,n]} \sum_{l \in [0,n]} \sum_j \tfrac{1}{g_j(k,l)} \left\{ a_j [\mathbf{p}_k^1 \cdot \mathbf{q}_l^1] + b_j [\mathbf{p}_k^0 \cdot \mathbf{q}_l^0] + f_j(k,l) r_k s_l \right\} - \langle \mathbf{Q}^\intercal \mathbf{1}, \Psi^\intercal \theta \rangle \right] + \langle \mathbf{y}, \Psi^\intercal \theta \rangle \right\}, \tag{42}$$

where Δ is the set of valid marginal probability matrix denoted as:

$$\Delta = \left\{ \mathbf{P} \left| \begin{array}{ll} p_{i,k} \geq 0 & \forall i, k \in [1, n] \\ p_{i,k} \leq \frac{1}{k} \sum_j p_{j,k} & \forall i, k \in [1, n] \\ \sum_k \frac{1}{k} \sum_i p_{i,k} \leq 1 & \end{array} \right. \right\}. \tag{43}$$

□

A.3 Proof of Theorem 3

Proof of Theorem 3. The result follows directly from the rule of subgradient of maximum function.

$$-\mathcal{L}(\theta) = \max_{\mathbf{Q} \in \Delta} \min_{\mathbf{P} \in \Delta} \left[- \sum_{k \in [0,n]} \sum_{l \in [0,n]} \sum_j \tfrac{1}{g_j(k,l)} \left\{ a_j [\mathbf{p}_k^1 \cdot \mathbf{q}_l^1] + b_j [\mathbf{p}_k^0 \cdot \mathbf{q}_l^0] + f_j(k,l) r_k s_l \right\} + \langle \mathbf{Q}^\intercal \mathbf{1}, \Psi^\intercal \theta \rangle \right] - \langle \mathbf{y}, \Psi^\intercal \theta \rangle \tag{44}$$

$$\partial_\theta - \mathcal{L}(\theta) \ni \Psi \left(\mathbf{Q}^{*\intercal} \mathbf{1} - \mathbf{y} \right), \text{ where:} \tag{45}$$

$$\mathbf{Q}^* = \operatorname*{argmax}_{\mathbf{Q} \in \Delta} \min_{\mathbf{P} \in \Delta} \left[- \sum_{k \in [0,n]} \sum_{l \in [0,n]} \sum_j \tfrac{1}{g_j(k,l)} \left\{ a_j [\mathbf{p}_k^1 \cdot \mathbf{q}_l^1] + b_j [\mathbf{p}_k^0 \cdot \mathbf{q}_l^0] + f_j(k,l) r_k s_l \right\} + \langle \mathbf{Q}^\intercal \mathbf{1}, \Psi^\intercal \theta \rangle \right]$$

□

A.4 Proof of Theorem 4

Proof of Theorem 4. The inner minimization over $\mathbf{Q}$ in Eq. (9) is:

$$\min_{\mathbf{Q}\in\Delta}\max_{\mathbf{P}\in\Delta}\left[\sum_{k\in[0,n]}\sum_{l\in[0,n]}\sum_{j}\tfrac{1}{g_j(k,l)}\left\{a_j[\mathbf{p}_k^1\cdot\mathbf{q}_l^1]+b_j[\mathbf{p}_k^0\cdot\mathbf{q}_l^0]+f_j(k,l)r_k s_l\right\}-\langle\mathbf{Q}^\intercal\mathbf{1},\Psi^\intercal\theta\rangle\right]. \tag{46}$$

Denote:

$$\mathcal{O}(\mathbf{Q},\mathbf{P})=\sum_{k\in[0,n]}\sum_{l\in[0,n]}\sum_{j}\tfrac{1}{g_j(k,l)}\left\{a_j[\mathbf{p}_k^1\cdot\mathbf{q}_l^1]+b_j[\mathbf{p}_k^0\cdot\mathbf{q}_l^0]+f_j(k,l)r_k s_l\right\}. \tag{47}$$

Since the objective in $\mathcal{O}(\mathbf{Q},\mathbf{P})$ is a bilinear function over $\mathbf{Q}$ and $\mathbf{P}$, it can be written in the form of $\mathcal{O}(\mathbf{Q},\mathbf{P})=\left\langle\frac{\partial\mathcal{O}(\mathbf{Q},\mathbf{P})}{\partial\mathbf{P}},\mathbf{P}\right\rangle+c(\mathbf{Q})$, where $c(\mathbf{Q})$ is the terms that are constant over $\mathbf{P}$. Therefore, Eq. (46) can be written as:

$$\min_{\mathbf{Q}\in\Delta}\max_{\mathbf{P}\in\Delta}\ \langle\mathbf{Z}(\mathbf{Q}),\mathbf{P}\rangle+c(\mathbf{Q})-\langle\mathbf{Q},\mathbf{W}\rangle, \tag{48}$$

where $\mathbf{Z}(\mathbf{Q})=\frac{\partial\mathcal{O}(\mathbf{Q},\mathbf{P})}{\partial\mathbf{P}}$, and $\mathbf{W}=\Psi^\intercal\theta\mathbf{1}^\intercal$. Note that both $\mathbf{Z}(\mathbf{Q})$ and $c(\mathbf{Q})$ are some linear functions that depend on the metric.

We expand the constraints over $\mathbf{P}$ as:

$$\begin{aligned}&\min_{\mathbf{Q}\in\Delta}\max_{\mathbf{P}}\ \langle\mathbf{Z}(\mathbf{Q}),\mathbf{P}\rangle+c(\mathbf{Q})-\langle\mathbf{Q},\mathbf{W}\rangle\\ \text{s.t.: }&p_{i,k}\geq 0\quad\forall i,k\in[1,n]\\ &p_{i,k}\leq\tfrac{1}{k}\textstyle\sum_j p_{j,k}\quad\forall i,k\in[1,n]\\ &\textstyle\sum_k\frac{1}{k}\sum_i p_{i,k}\leq 1\end{aligned} \tag{49}$$

We now perform a change of variable. Let us transform $\mathbf{P}$ to a matrix $\mathbf{A}$ where its element contains the value of $a_{i,k}=\frac{1}{k}a_{i,k}$. We can rewrite the objective as:

$$\begin{aligned}&\min_{\mathbf{Q}\in\Delta}\max_{\mathbf{A}}\ \langle\mathbf{Z}'(\mathbf{Q}),\mathbf{A}\rangle+c(\mathbf{Q})-\langle\mathbf{Q},\mathbf{W}\rangle\\ \text{s.t.: }&a_{i,k}\geq 0\quad\forall i,k\in[1,n]\\ &a_{i,k}\leq\tfrac{1}{k}\textstyle\sum_j a_{j,k}\quad\forall i,k\in[1,n]\\ &\textstyle\sum_k\sum_i a_{i,k}\leq 1,\end{aligned} \tag{50}$$

where $\mathbf{Z}'(\mathbf{Q})$ is the linearly transformed $\mathbf{Z}(\mathbf{Q})$ to adjust the transformation of the variable from $\mathbf{P}$ to $\mathbf{A}$.

Using duality, we introduce a Lagrange dual variable for $a_{i,k}\leq\frac{1}{k}\sum_j a_{j,k}$ constraint.

$$\begin{aligned}&\min_{\mathbf{Q}\in\Delta;\boldsymbol{\alpha}\geq 0}\max_{\mathbf{A}}\ \langle\mathbf{Z}'(\mathbf{Q}),\mathbf{A}\rangle+c(\mathbf{Q})-\langle\mathbf{Q},\mathbf{W}\rangle-\sum_{i,k}\alpha_{i,k}\left(a_{ik}-\tfrac{1}{k}\textstyle\sum_j a_{j,k}\right)\\ \text{s.t.: }&a_{i,k}\geq 0\quad\forall i,k\in[1,n]\\ &\textstyle\sum_k\sum_i a_{i,k}\leq 1\end{aligned} \tag{51}$$

We regroup the terms that depend on $\mathbf{A}$ as:

$$\begin{aligned}&\min_{\mathbf{Q}\in\Delta;\boldsymbol{\alpha}\geq 0}\max_{\mathbf{A}}\ \langle\mathbf{Z}'(\mathbf{Q}),\mathbf{A}\rangle-\sum_{i,k}a_{i,k}\left(\alpha_{ik}-\tfrac{1}{k}\textstyle\sum_j\alpha_{j,k}\right)+c(\mathbf{Q})-\langle\mathbf{Q},\mathbf{W}\rangle\\ \text{s.t.: }&a_{i,k}\geq 0\quad\forall i,k\in[1,n]\\ &\textstyle\sum_k\sum_i a_{i,k}\leq 1\end{aligned} \tag{52}$$

We now eliminate the inner maximization over $\mathbf{A}$ by transforming it into constraints as follows:

$$\begin{aligned}&\min_{\mathbf{Q}\in\Delta;\boldsymbol{\alpha}\geq 0;v}\ v+c(\mathbf{Q})-\langle\mathbf{Q},\mathbf{W}\rangle\\ \text{s.t.: }&v\geq 0\\ &v\geq(\mathbf{Z}'(\mathbf{Q}))_{(i,k)}-\alpha_{i,k}+\tfrac{1}{k}\textstyle\sum_j\alpha_{j,k},\qquad\forall i,k\in[1,n].\end{aligned} \tag{53}$$

The formulation above can be written in a standard linear program as:

$$\begin{aligned}
&\min_{\mathbf{Q};\boldsymbol{\alpha};v} \; v + c(\mathbf{Q}) - \langle \mathbf{Q}, \mathbf{W} \rangle \\
\text{s.t.:}\; & q_{i,k} \geq 0 \quad \forall i,k \in [1,n] \\
& \alpha_{i,k} \geq 0 \quad \forall i,k \in [1,n] \\
& v \geq 0 \\
& q_{i,k} \leq \tfrac{1}{k} \textstyle\sum_j q_{j,k} \quad \forall i,k \in [1,n] \\
& \textstyle\sum_k \tfrac{1}{k} \sum_i q_{i,k} \leq 1 \\
& v \geq (\mathbf{Z}'(\mathbf{Q}))_{(i,k)} - \alpha_{i,k} + \tfrac{1}{k} \textstyle\sum_j \alpha_{j,k}, \qquad \forall i,k \in [1,n],
\end{aligned} \tag{54}$$

where $c(\mathbf{Q})$ is a linear function of $\mathbf{Q}$ and $\mathbf{Z}'(\mathbf{Q})$ is a matrix-valued linear function of $\mathbf{Q}$, both of which are defined analytically by the form of the metric. □

A.5 Proof of Theorem 5

Proof of Theorem 5. Let us take a look at the expectation in the constraints:

$$\mathbb{E}_{\mathcal{P}(\hat{\mathbf{Y}})} \Big[\text{metric}(\hat{\mathbf{Y}}, \mathbf{Y}) \Big] \tag{55}$$

$$= \sum_{\hat{\mathbf{y}} \in \{0,1\}^n} \mathcal{P}(\hat{\mathbf{y}}) \; \text{metric}(\hat{\mathbf{y}}, \mathbf{y}) \tag{56}$$

$$= \sum_{\hat{\mathbf{y}} \in \{0,1\}^n} \mathcal{P}(\hat{\mathbf{y}}) \sum_j \frac{a_j \sum_i \hat{y}_i y_i + b_j \sum_i (1-\hat{y}_i)(1-y_i) + f_j(\sum_i \hat{y}_i, \sum_i y_i)}{g_j(\sum_i \hat{y}_i, \; \sum_i y_i)} \tag{57}$$

$$= \sum_{k \in [0,n]} \sum_j \frac{a_j \sum_{\{\hat{\mathbf{y}} | \Sigma_i \hat{y}_i = k\}} \mathcal{P}(\hat{\mathbf{y}}) \sum_i \hat{y}_i y_i + b_j \sum_{\{\hat{\mathbf{y}} | \Sigma_i \hat{y}_i = k\}} \mathcal{P}(\hat{\mathbf{y}}) \sum_i (1-\hat{y}_i)(1-y_i) + \sum_{\{\hat{\mathbf{y}} | \Sigma_i \hat{y}_i = k\}} \mathcal{P}(\hat{\mathbf{y}}) f_j(k,l)}{g_j(k,l)} \tag{58}$$

$$= \sum_{k \in [0,n]} \sum_j \frac{a_j \sum_i \mathcal{P}(\hat{y}_i = 1, \sum_{i'} \hat{y}_{i'} = k) y_i + b_j \sum_i \mathcal{P}(\hat{y}_i = 0, \sum_{i'} \hat{y}_{i'} = k)(1-y_i) + \sum_i \mathcal{P}(\sum_i \hat{y}_i = k) f_j(k,l)}{g_j(k,l)} \tag{59}$$

$$= \sum_{k \in [0,n]} \sum_j \frac{a_j [\mathbf{p}_k^1 \cdot \mathbf{y}] + b_j [\mathbf{p}_k^0 \cdot (1-\mathbf{y})] + f_j(k,l) r_k}{g_j(k,l)} \tag{60}$$

where $l = \sum_i y_i$. Therefore, the metric constraints can be written as:

$$\sum_{k \in [0,n]} \sum_j \frac{a_j [\mathbf{p}_k^1 \cdot \mathbf{y}] + b_j [\mathbf{p}_k^0 \cdot (1-\mathbf{y})] + f_j(k,l) r_k}{g_j(k,l)} \geq \tau_i, \; \forall i \in [1,t]$$

The dual formulation of Eq. (13) is:

$$\begin{aligned}
&\max_{\theta} \mathbb{E}_{\tilde{P}(\mathbf{X},\mathbf{Y})} \left[\min_{\mathcal{Q}(\check{\mathbf{Y}})} \max_{\mathcal{P}(\hat{\mathbf{Y}}) \in \Gamma} \mathbb{E}_{\mathcal{P}(\hat{\mathbf{Y}});\mathcal{Q}(\check{\mathbf{Y}})} \left[\text{metric}^{(0)}(\hat{\mathbf{Y}}, \check{\mathbf{Y}}) + \theta^\intercal \left(\phi(\mathbf{X}, \check{\mathbf{Y}}) - \phi(\mathbf{X}, \mathbf{Y}) \right) \right] \right] \\
&\text{where} : \; \Gamma \triangleq \left\{ \mathcal{P}(\hat{\mathbf{Y}}) \mid \mathbb{E}_{\tilde{P}(\mathbf{X},\mathbf{Y});\mathcal{P}(\hat{\mathbf{Y}})} \left[\text{metric}^{(i)}(\hat{\mathbf{Y}}, \mathbf{Y}) \right] \geq \tau_i, \; \forall i \in [1,t] \right\}.
\end{aligned} \tag{61}$$

Following the analysis in the proof of Theorem 2, the dual formulation can be simplified as:

$$\max_{\theta} \left\{ \min_{\mathbf{Q} \in \Delta} \max_{\mathbf{P} \in \Delta \cap \Gamma} \left[\sum_{k \in [0,n]} \sum_{l \in [0,n]} \sum_j \tfrac{1}{g_j^{(0)}(k,l)} \left\{ a_j^{(0)} [\mathbf{p}_k^1 \cdot \mathbf{q}_l^1] + b_j^{(0)} [\mathbf{p}_k^0 \cdot \mathbf{q}_l^0] + f_j^{(0)}(k,l) r_k s_l \right\} - \langle \mathbf{Q}^\intercal \mathbf{1}, \Psi^\intercal \theta \rangle \right] + \langle \mathbf{y}, \Psi^\intercal \theta \rangle \right\},$$

where:

$$\Delta = \left\{ \mathbf{P} \;\middle|\; \begin{array}{ll} p_{i,k} \geq 0 & \forall i,k \in [1,n] \\ p_{i,k} \leq \frac{1}{k} \sum_j p_{j,k} & \forall i,k \in [1,n] \\ \sum_k \frac{1}{k} \sum_i p_{i,k} \leq 1 & \end{array} \right\}, \text{ and} \tag{62}$$

$$\Gamma = \left\{ \mathbf{P} \middle| \sum_{k\in[0,n]} \sum_j \frac{a_j^{(i)}[\mathbf{p}_k^1 \cdot \mathbf{y}] + b_j^{(i)}[\mathbf{p}_k^0 \cdot (1-\mathbf{y})] + f_j^{(i)}(k,l) r_k)}{g_j^{(i)}(k,l)} \geq \tau_i,\ \forall i \in [1,t] \right\},\ \text{where } l = \textstyle\sum_{i'} y_{i'}. \tag{63}$$

□

A.6 Proof of Theorem 6

Proof of Theorem 6. The inner minimization over $\mathbf{Q}$ in Eq. (14) is relatively similar to the standard case (Eq. (9)). The only difference is the additional constraints over $\mathbf{P}$. Since the numerators of the metrics in the constraints are linear in terms of $\mathbf{p}_k^1$ and $\mathbf{p}_k^0$ (which also means linear in terms of $\mathbf{p}_k$), then the constraints in Γ can be represented by some matrix $\mathbf{B}^{(i)}$ and some constant μ_i such that:

$$\langle \mathbf{B}^{(i)}, \mathbf{P} \rangle + \mu_i \geq \tau_i, \qquad \text{or,} \qquad \textstyle\sum_k (\mathbf{b}_k^{(i)})^\intercal \mathbf{p}_k^{(i)} + \mu_i \geq \tau_i, \qquad \forall i \in [1,t] \tag{64}$$

Following the change of variable in the proof of Theorem 4, we can also represent the constraint in terms of $\mathbf{A}$ using some matrix $\mathbf{B}'^{(i)}$ such that:

$$\langle \mathbf{B}'^{(i)}, \mathbf{A} \rangle + \mu_i \geq \tau_i, \qquad \text{or,} \qquad \textstyle\sum_k (\mathbf{b}_k'^{(i)})^\intercal \mathbf{a}_k^{(i)} + \mu_i \geq \tau_i, \qquad \forall i \in [1,t] \tag{65}$$

Therefore, we have an inner optimization over $\mathbf{Q}$ and $\mathbf{A}$, which can be written as:

$$\begin{aligned} &\min_{\mathbf{Q}\in\Delta} \max_{\mathbf{A}}\ \langle \mathbf{Z}'(\mathbf{Q}), \mathbf{A} \rangle + c(\mathbf{Q}) - \langle \mathbf{Q}, \mathbf{W} \rangle \\ \text{s.t.:}\ & a_{i,k} \geq 0 \quad \forall i,k \in [1,n] \\ & a_{i,k} \leq \tfrac{1}{k} \textstyle\sum_j a_{j,k} \quad \forall i,k \in [1,n] \\ & \textstyle\sum_k \sum_i a_{i,k} \leq 1 \\ & \langle \mathbf{B}'^{(l)}, \mathbf{A} \rangle + \mu_l \geq \tau_l, \forall l \in [1,t] \end{aligned} \tag{66}$$

Using duality, we introduce Lagrange dual variables.

$$\begin{aligned} &\min_{\mathbf{Q}\in\Delta;\boldsymbol{\alpha}\geq 0;\boldsymbol{\beta}\geq 0} \max_{\mathbf{A}}\ \langle \mathbf{Z}'(\mathbf{Q}), \mathbf{A} \rangle + c - \langle \mathbf{Q}, \mathbf{W} \rangle - \sum_{i,k} \alpha_{i,k} \left(a_{ik} - \tfrac{1}{k} \textstyle\sum_j a_{j,k} \right) + \sum_l \beta_l \left(\langle \mathbf{B}'^{(l)}, \mathbf{A} \rangle + \mu_l - \tau_l \right) \\ \text{s.t.:}\ & a_{i,k} \geq 0 \quad \forall i,k \in [1,n] \\ & \textstyle\sum_k \sum_i a_{i,k} \leq 1 \end{aligned} \tag{67}$$

We can convert the optimization in a standard linear program format as follows:

$$\begin{aligned} &\min_{\mathbf{Q};\boldsymbol{\alpha};\boldsymbol{\beta};v}\ v + c(\mathbf{Q}) - \langle \mathbf{Q}, \mathbf{W} \rangle + \sum_l (\mu_l - \tau_l) \\ \text{s.t.:}\ & q_{i,k} \geq 0 \quad \forall i,k \in [1,n] \\ & \alpha_{i,k} \geq 0 \quad \forall i,k \in [1,n] \\ & \beta_l \geq 0 \quad \forall l \in [1,s] \\ & v \geq 0 \\ & q_{i,k} \leq \tfrac{1}{k} \textstyle\sum_j q_{j,k} \quad \forall i,k \in [1,n] \\ & \textstyle\sum_k \tfrac{1}{k} \sum_i q_{i,k} \leq 1 \\ & v \geq (\mathbf{Z}'(\mathbf{Q}))_{(i,k)} - \alpha_{i,k} + \tfrac{1}{k} \textstyle\sum_j \alpha_{j,k} + \sum_l \beta_l \, (\mathbf{B}'^{(l)})_{(i,k)}, \qquad \forall i,k \in [1,n]. \end{aligned} \tag{68}$$

□

A.7 Proof of Theorem 7

Proof of Theorem 7. Despite its apparent differences from standard empirical risk minimization (ERM), the dual formulation of the adversarial prediction (Eq. (2)) can be equivalently recast as an ERM method:

$$\min_{\theta} \mathbb{E}_{\tilde{P}(\mathbf{X},\mathbf{Y})} \left[AL_{h_\theta}(\mathbf{X},\mathbf{Y}) \right], \qquad \text{where:} \tag{69}$$

$$AL_{h_\theta}(\mathbf{X},\mathbf{Y}) = \max_{\mathcal{Q}(\check{\mathbf{Y}})} \min_{\mathcal{P}(\hat{\mathbf{Y}})} \mathbb{E}_{\mathcal{P}(\hat{\mathbf{Y}});\mathcal{Q}(\check{\mathbf{Y}})} \Big[-\text{metric}(\hat{\mathbf{Y}},\check{\mathbf{Y}}) + h_\theta(\mathbf{X},\check{\mathbf{Y}}) - h_\theta(\mathbf{X},\mathbf{Y}) \Big] \tag{70}$$

and $h_\theta(\mathbf{x},\mathbf{y}) = \theta^\intercal \phi(\mathbf{x},\mathbf{y})$ is the Lagrangian potential function. $AL_{h_\theta}(\mathbf{x},\mathbf{y})$ is then the surrogate loss for input $\mathbf{x}$ and label $\mathbf{y}$. The Fisher consistency condition for the adversarial prediction can then be written as:

$$\begin{aligned} & h^* \in \mathcal{H}^* \triangleq \operatorname*{argmin}_f \mathbb{E}_{P(\mathbf{Y}|\mathbf{x})} \left[\text{AL}_h(\mathbf{x},\mathbf{Y}) \right] \\ \Rightarrow \ & \operatorname*{argmax}_{\mathbf{y}} h^*(\mathbf{x},\mathbf{y}) \subseteq \operatorname*{argmax}_{\mathbf{y}'} \mathbb{E}_{P(\mathbf{Y}|\mathbf{x})} [\text{metric}(\mathbf{y}',\mathbf{Y})]. \end{aligned} \tag{71}$$

It has been shown by Fathony et al. (2018a,b), for a given natural requirement of performance metric, i.e., $\text{metric}(\mathbf{y},\mathbf{y}) > \text{metric}(\mathbf{y},\mathbf{y}')$ for all $\mathbf{y}' \neq \mathbf{y}$, the adversarial prediction is Fisher consistent provided that h is optimized over all measurable functions over the input space of $(\mathbf{x},\mathbf{y})$. We quote the result below:

Proposition 1 (Consistency result from Fathony et al. (2018a,b))**.** *Suppose we have a metric that satisfy the natural requirement: metric*$(\mathbf{y},\mathbf{y}) >$ *metric*$(\mathbf{y},\mathbf{y}')$ *for all* $\mathbf{y}' \neq \mathbf{y}$*. Then the adversarial surrogate loss* AL_h *is Fisher consistent if* h *is optimized over all measurable functions over the input space of* $(\mathbf{x},\mathbf{y})$*.*

The key to the result above is the observation that given a loss metric $\text{loss}(\mathbf{y}',\mathbf{y})$, for the optimal potential function h^*, $h^*(\mathbf{x},\mathbf{y}) + \text{loss}(\mathbf{y}^\diamond,\mathbf{y})$ is invariant to $\mathbf{y}$ where $y^\diamond = \text{argmax}_{\mathbf{y}'} \mathbb{E}_{P(\mathbf{Y}|\mathbf{x})}[\text{metric}(\mathbf{y}',\mathbf{Y})]$. This property is referred to as the *loss reflective* property of the h minimizer. For a performance metric, the property can be equivalently written as $h^*(\mathbf{x},\mathbf{y}) - \text{metric}(\mathbf{y}^\diamond,\mathbf{y})$ is invariant to $\mathbf{y}$.

We now want to reduce the input space that h needs to operate in order to achieve to Fisher consistency property. We consider the restricted set of h defined as: $h(\mathbf{x},\mathbf{y}) = \sum_{i,k} \rho_{i,k}(\mathbf{x},y_i,k)\mathbb{I}[\sum_i y_i = k]$, where each $\rho_{\{i,k\}}$ is optimized over the set of all measurable functions on the individual input space of $(\mathbf{x},y_i)$. If the performance metric follows the construction in Eq. (3), then we can achieve the loss reflective property under the restricted set of h by setting:

$$\rho_{i,k}(\mathbf{x},y_i,k) = \sum_j \frac{a_j \sum_i y_i^\diamond y_i + b_j \sum_i (1-y_i^\diamond)(1-y_i) + f_j(\sum_i y_i^\diamond, k)}{g_j(\sum_i y_i^\diamond, k)}. \tag{72}$$

This will render the loss reflective property as $h^*(\mathbf{x},\mathbf{y}) - \text{metric}(\mathbf{y}^\diamond,\mathbf{y}) = \mathbf{0}$.

Therefore, we can conclude that our method is Fisher consistent for a performance metric that follows the construction in Eq. (3) if the algorithm is optimized over a set of functions that are additive over each sample and sum statistics. □

Appendix B Experiment Details

To evaluate our approach, we apply our formulation to classification tasks on 20 different tabular datasets from the UCI repository (Dua and Graff, 2017) and benchmark datasets (Chu and Ghahramani, 2005), as well as image datasets from MNIST and Fashion MNIST. Table 3 shows the list of the datasets and their properties (the number of samples in the train, validation, and test sets). Some of the datasets are binary classification tasks, which we use directly in our experiments. For the multiclass datasets, we transform them into binary classification tasks by selecting one or more classes as the positive label and the rest as the negative label. Table 3 also shows the original class labels in the dataset and the classes that we select as the positive label in the transformed binary classification. The distribution of the positive and negative samples in the training set of the resulting binary classification tasks is described in Table 4. For all of the datasets, we perform standardization, i.e., transform all the variables into zero mean and unit variance. For the datasets that have not been divided into training and testing set, we split the data with the rule of 70% samples for the train set and 30% for the test set. In addition, during the training, we also split the original training set into two different sets, 80% of the set for training, and the rest 20% of the set for validation.

Table 3: Properties of the datasets used in the experiments

Dataset	# train set	# validation set	# test set	original classes	positive classes
abalone	2,338	585	1,254	[1,10]	[6,10]
adult	25,324	6,331	13,567	[0,1]	[1]
appliancesenergy	11,051	2,763	5,921	[0,1]	[1]
bankdomains2	4,587	1,147	2,458	[1,10]	[7,10]
bankmarketing	25,318	6,329	13,564	[0,1]	[1]
californiahousing	11,558	2,889	6,193	[1,10]	[7,10]
censusdomains	12,758	3,190	6,836	[1,10]	[7,10]
computeractivity2	4,587	1,147	2,458	[1,10]	[8,10]
default	16,800	4,200	9,000	[0,1]	[1]
dutch	33,835	8,459	18,126	[0,1]	[1]
eegeye	8,389	2,097	4,494	[0,1]	[1]
fashion-mnist	48,000	12,000	10,000	[0,9]	[0]
htru2	10,022	2,506	5,370	[0,1]	[1]
letter	11,200	2,800	6,000	[1,26]	[22,26]
mnist	48,000	12,000	10,000	[0,9]	[0]
onlinenews	22,200	5,550	11,894	[0,1]	[1]
pageblocks	3,065	766	1,642	[1,5]	[4,5]
redwine	895	224	480	[1,10]	[7,10]
sat	3,548	887	2,000	[1,7]	[6,7]
sensorless	32,765	8,191	17,553	[1,11]	[7,10]
shuttle	34,800	8,700	14,500	[1,7]	[4,7]
whitewine	2,743	686	1,469	[1,10]	[7,10]

For the tabular datasets, we construct a multi-layer perceptron (MLP) with two hidden layers. Each layer has 100 nodes. For the image datasets, we construct a convolutional neural network (CNN) with two convolutional layers and two dense layers. In the training process, we use the standard gradient descent algorithm for both the BCE and AP-Perf networks. We use the learning rate of 0.01 for the BCE networks and 0.003 for the AP-Perf networks. We select the learning rate values for both methods based on the training and validation test performance plot over 100 epochs.

For both methods, we perform a cross-validation using validation set to select the best L2 regularization among $\lambda = \{0, 0.001, 0.01, 0.1\}$. After the training session finished, we compute the value of the metric for prediction in the testing dataset. For both methods, we select the predictive models that achieve the best metric in the validation set. We also implement an early stopping technique based on the validation set to avoid overfitting. Even though we run all the networks for 100 epochs, we select the parameters on the epoch that produce the best metric on the validation set. We then use this parameter to make predictions on the testing set.

Appendix C Code Examples for Constructing Performance Metrics

C.1 Commonly Used Performance Metrics

Below are some code examples for constructing some of commonly used performance metrics.

```
@metric Accuracy      # Accuracy
function define(::Type{Accuracy}, C::ConfusionMatrix)
    return (C.tp + C.tn) / (C.all)
end

accuracy_metric = Accuracy()
```

Table 4: The number of positive and negative samples in the train set for each dataset

Dataset	# train set	# positive	# negative	positive percentage
abalone	2338	146	2192	6%
adult	25324	6258	19066	25%
appliancesenergy	11051	2961	8090	27%
bankdomains2	4587	1829	2758	40%
bankmarketing	25318	2941	22377	12%
californiahousing	11558	4637	6921	40%
censusdomains	12758	5088	7670	40%
computeractivity2	4587	1379	3208	30%
default	16800	3701	13099	22%
dutch	33835	17803	16032	53%
eegeye	8389	3769	4620	45%
fashion-mnist	48000	4764	43236	10%
htru2	10022	901	9121	9%
letter	11200	2167	9033	19%
mnist	48000	4729	43271	10%
onlinenews	22200	2899	19301	13%
pageblocks	3065	118	2947	4%
redwine	895	113	782	13%
sat	3548	819	2729	23%
sensorless	32765	11934	20831	36%
shuttle	34800	7408	27392	21%
whitewine	2743	587	2156	21%

```
@metric Precision       # Precision
function define(::Type{Precision}, C::ConfusionMatrix)
    return C.tp / C.pp
end

prec = Precision()
special_case_positive!(prec)


@metric Recall       # Recall / Sensitivity
function define(::Type{Recall}, C::ConfusionMatrix)
    return C.tp / C.ap
end

rec = Recall()
special_case_positive!(rec)


@metric Specificity       # Specificity
function define(::Type{Specificity}, C::ConfusionMatrix)
    return C.tn / C.an
end
spec = Specificity()
special_case_negative!(spec)


@metric F1Score         # F1 Score
function define(::Type{F1Score}, C::ConfusionMatrix)
    return (2 * C.tp) / (C.ap + C.pp)
end

f1_score = F1Score()
special_case_positive!(f1_score)
```

```
@metric GM_PrecRec      # Geometric Mean of Prec and Rec
function define(::Type{GM_PrecRec}, C::ConfusionMatrix)
    return C.tp / sqrt(C.ap * C.pp)
end

gpr = GM_PrecRec()
special_case_positive!(gpr)


@metric Informedness     # informedness
function define(::Type{Informedness}, C::ConfusionMatrix)
    return C.tp / C.ap + C.tn / C.an - 1
end

inform = Informedness()
special_case_positive!(inform)
special_case_negative!(inform)


@metric Kappa       # Cohen's kappa score
function define(::Type{Kappa}, C::ConfusionMatrix)
    num = (C.tp + C.tn) / C.all - (C.ap * C.pp + C.an * C.pn) / C.all^2
    den = 1 - (C.ap * C.pp + C.an * C.pn) / C.all^2
    return num / den
end

kappa = Kappa()
special_case_positive!(kappa)
special_case_negative!(kappa)


@metric PrecisionGvRecall         # precision given recall >= 0.8
function define(::Type{PrecisionGvRecall}, C::ConfusionMatrix)
    return C.tp / C.pp
end
function constraint(::Type{PrecisionGvRecall}, C::ConfusionMatrix)
    return C.tp / C.ap >= 0.8
end

precision_gv_recall = PrecisionGvRecall()
special_case_positive!(precision_gv_recall)
cs_special_case_positive!(precision_gv_recall, true)


@metric RecallGvPrecision         # recall given precision
function define(::Type{RecallGvPrecision}, C::ConfusionMatrix)
    return C.tp / C.pp
end
function constraint(::Type{RecallGvPrecision}, C::ConfusionMatrix)
    return C.tp / C.ap >= 0.8
end

recal_gv_precision = RecallGvPrecision()
special_case_positive!(recal_gv_precision)
cs_special_case_positive!(recal_gv_precision, true)


@metric PrecisionGvRecallSpecificity         # precision given recall >= 0.8 and specificity >= 0.8
function define(::Type{PrecisionGvRecallSpecificity}, C::ConfusionMatrix)
    return C.tp / C.pp
end
function constraint(::Type{PrecisionGvRecallSpecificity}, C::ConfusionMatrix)
    return [C.tp / C.ap >= 0.8,
            C.tn / C.an >= 0.8]
end
```

```
precision_gv_recall_spec = PrecisionGvRecallSpecificity()
special_case_positive!(precision_gv_recall_spec)
cs_special_case_positive!(precision_gv_recall_spec, [true, false])
cs_special_case_negative!(precision_gv_recall_spec, [false, true])
```

C.2 Performance Metrics with Arguments

Our framework also supports writing performance metric with arguments, for example, the F_β score metric which depends on the value of β. Below are some examples on constructing metrics with arguments.

```
@metric FBeta beta          # F-Beta
function define(::Type{FBeta}, C::ConfusionMatrix, beta)
    return ((1 + beta^2) * C.tp) / (beta^2 * C.ap + C.pp)
end

f1_score = FBeta(1)
special_case_positive!(f1_score)

f2_score = FBeta(2)
special_case_positive!(f2_score)


# precision given recall
@metric PrecisionGvRecall th
function define(::Type{PrecisionGvRecall}, C::ConfusionMatrix, th)
    return C.tp / C.pp
end

function constraint(::Type{PrecisionGvRecall}, C::ConfusionMatrix, th)
    return C.tp / C.ap >= th
end

precision_gv_recall_80 = PrecisionGvRecall(0.8)
special_case_positive!(precision_gv_recall_80)
cs_special_case_positive!(precision_gv_recall_80, true)

precision_gv_recall_60 = PrecisionGvRecall(0.6)
special_case_positive!(precision_gv_recall_60)
cs_special_case_positive!(precision_gv_recall_60, true)

precision_gv_recall_95 = PrecisionGvRecall(0.95)
special_case_positive!(precision_gv_recall_95)
cs_special_case_positive!(precision_gv_recall_95, true)


@metric PrecisionGvRecallSpecificity th1 th2        # precision given recall >= th1 and specificity >= th2
function define(::Type{PrecisionGvRecallSpecificity}, C::ConfusionMatrix, th1, th2)
    return C.tp / C.pp
end
function constraint(::Type{PrecisionGvRecallSpecificity}, C::ConfusionMatrix, th1, th2)
    return [C.tp / C.ap >= th1,
            C.tn / C.an >= th2]
end

precision_gv_recall_spec = PrecisionGvRecallSpecificity(0.8, 0.8)
special_case_positive!(precision_gv_recall_spec)
cs_special_case_positive!(precision_gv_recall_spec, [true, false])
cs_special_case_negative!(precision_gv_recall_spec, [false, true])
```

Appendix D Linear Program Solver using the ADMM Technique

In this section we construct an ADMM formulation for solving the inner optimization over $\mathbf{Q}$ in Eq. (9). The optimization can also be solved using any linear program solver as shown in the Appendix A.4. However, the

runtime complexity of solving the LP is $O(m^6)$ where m is the batch size, which makes it impractical for a batch of size greater than 30 samples. Our ADMM formulation reduces the runtime complexity to $O(m^3)$.

We consider an extension of the family of evaluation metrics in Eq. (3) to also include the false positive and the false negative in the numerator of the fractions, i.e.,

$$\text{metric}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_j \frac{a_j \mathsf{TP} + b_j \mathsf{TN} + c_j \mathsf{FP} + d_j \mathsf{FN} + f_j(\mathsf{PP}, \mathsf{AP})}{g_j(\mathsf{PP}, \mathsf{AP})}, \tag{73}$$

where a_j, b_j, c_j, and d_j are constants.

D.1 ADMM Formulation for Metrics with the Special Case for True Positive

We start with a task where the metric enforces a special case for true positive (for example, the precision, recall, and F1-score). In this task, the optimization over $\mathbf{Q}$ in Eq. (9) becomes:

$$\min_{\mathbf{Q}\in\Delta} \max_{\mathbf{P}\in\Delta} \Bigg[\sum_{k,l\in[1,n]} \sum_j \tfrac{1}{g_j(k,l)} \Big\{ a_j[\mathbf{p}_k^1 \cdot \mathbf{q}_l^1] + b_j[\mathbf{p}_k^0 \cdot \mathbf{q}_l^0] + c_j[\mathbf{p}_k^1 \cdot \mathbf{q}_l^0] + d_j[\mathbf{p}_k^0 \cdot \mathbf{q}_l^1] + f_j(k,l) r_k s_l \Big\} + \mathcal{P}(\mathbf{0})\mathcal{Q}(\mathbf{0}) - \langle \mathbf{Q}^\intercal \mathbf{1}, \Psi^\intercal \theta \rangle \Bigg]. \tag{74}$$

In this section we will use matrix notations in our formulation, extending our vector notations in Appendix A.2. Using matrix notations, Eq. (74) can be written as:

$$\min_{\{\mathbf{Q}_1, \mathbf{Q}_0, \mathbf{s}, v_0\}\in\Delta} \max_{\{\mathbf{P}_1, \mathbf{P}_0, \mathbf{r}, u_0\}\in\Delta} \langle \mathbf{M}_1, \mathbf{P}_1^\intercal \mathbf{Q}_1 \rangle + \langle \mathbf{M}_2, \mathbf{P}_1^\intercal \mathbf{Q}_0 \rangle + \langle \mathbf{M}_3, \mathbf{P}_0^\intercal \mathbf{Q}_1 \rangle + \langle \mathbf{M}_4, \mathbf{P}_0^\intercal \mathbf{Q}_0 \rangle + \langle \mathbf{M}_5, \mathbf{r}\mathbf{s}^\intercal \rangle + u_0 v_0 - \langle \mathbf{Q}_1, \mathbf{\Omega} \rangle, \tag{75}$$

where the matrix variables $\mathbf{Q}_1$, $\mathbf{Q}_0$, $\mathbf{P}_1$, and $\mathbf{P}_0$ represent:

$$\begin{aligned}
[\mathbf{Q}_1]_{i,j} &= \mathcal{Q}(\check{y}_i = 1, \textstyle\sum_l \check{y}_l = j), \quad i,j \in \{1,\dots,n\} \\
[\mathbf{Q}_0]_{i,j} &= \mathcal{Q}(\check{y}_i = 0, \textstyle\sum_l \check{y}_l = j), \quad i,j \in \{1,\dots,n\} \\
[\mathbf{P}_1]_{i,j} &= \mathcal{P}(\hat{y}_i = 1, \textstyle\sum_l \hat{y}_l = j), \quad i,j \in \{1,\dots,n\} \\
[\mathbf{P}_0]_{i,j} &= \mathcal{P}(\hat{y}_i = 0, \textstyle\sum_l \hat{y}_l = j), \quad i,j \in \{1,\dots,n\},
\end{aligned}$$

the vector and scalar variables represent:

$$\begin{aligned}
[\mathbf{s}]_j &= \mathcal{Q}(\textstyle\sum_l \check{y}_l = j), \quad j \in \{1,\dots,n\} \\
v_0 &= \mathcal{Q}(\textstyle\sum_l \check{y}_l = 0) \\
[\mathbf{r}]_j &= \mathcal{P}(\textstyle\sum_l \hat{y}_l = j), \quad j \in \{1,\dots,n\} \\
u_0 &= \mathcal{P}(\textstyle\sum_l \hat{y}_l = 0),
\end{aligned}$$

and the matrix $\mathbf{\Omega} = \Psi^\intercal \theta \mathbf{1}^\intercal$.

The matrix coefficients $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{M}_3$, $\mathbf{M}_4$, and $\mathbf{M}_5$ are computed from the performance metric, where each cell k,l of the matrices represents:

$$[\mathbf{M}_1]_{k,l} = \sum_j \frac{a_j}{g_j(k,l)}, \quad [\mathbf{M}_2]_{k,l} = \sum_j \frac{b_j}{g_j(k,l)}, \quad [\mathbf{M}_3]_{k,l} = \sum_j \frac{c_j}{g_j(k,l)},$$
$$[\mathbf{M}_4]_{k,l} = \sum_j \frac{d_j}{g_j(k,l)}, \quad [\mathbf{M}_5]_{k,l} = \sum_j \frac{f_j(k,l)}{g_j(k,l)}.$$

We write the original marginal distribution constraint Δ over $\mathbf{P}$ in matrix notations over $\{\mathbf{P}_1, \mathbf{P}_0, \mathbf{r}, u_0\}$ as:

$$\begin{aligned}
&\mathbf{P}_1 \geq 0,\ \mathbf{P}_0 \geq 0,\ \mathbf{r} \geq 0,\ u_0 \geq 0 \\
&\mathbf{r} = \text{diag}(\boldsymbol{\kappa}) \mathbf{P}_1^\intercal \mathbf{1} \\
&\mathbf{r}^\intercal \mathbf{1} + u_0 = 1 \\
&\mathbf{P}_1 + \mathbf{P}_0 = \mathbf{1}\mathbf{r}^\intercal,
\end{aligned}$$

where: $\boldsymbol{\kappa} = [\frac{1}{1}, \frac{1}{2}, \ldots, \frac{1}{n}]^\intercal$. All of the inequalities are element-wise.

Similarly, we write the original marginal distribution constraint Δ over $\mathbf{Q}$ in matrix notations over $\{\mathbf{Q}_1, \mathbf{Q}_0, \mathbf{s}, v_0\}$ as:

$$\begin{aligned}
&\mathbf{Q}_1 \geq 0,\ \mathbf{Q}_0 \geq 0,\ \mathbf{s} \geq 0,\ v_0 \geq 0 \\
&\mathbf{s} = \operatorname{diag}(\boldsymbol{\kappa})\mathbf{Q}_1^\intercal \mathbf{1} \\
&\mathbf{s}^\intercal \mathbf{1} + v_0 = 1 \\
&\mathbf{Q}_1 + \mathbf{Q}_0 = \mathbf{1}\mathbf{s}^\intercal.
\end{aligned}$$

D.1.1 Simplification and Reformulation

As mentioned in Appendix A.2, we can compute all the variables for $\mathcal{P}(y_i = 0, \ldots)$ from the variables for $\mathcal{P}(y_i = 1, \ldots)$. Specifically, we can derive $\mathbf{P}_0$, $\mathbf{r}$, and u_0 from $\mathbf{P}_1$. Let we denote $\mathbf{P} = \mathbf{P}_1$, then the equalities below hold:

$$\mathbf{P}_0 = \mathbf{1}\mathbf{1}^\intercal \mathbf{P} \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{P} \tag{76}$$
$$\mathbf{r} = \operatorname{diag}(\boldsymbol{\kappa})\mathbf{P}^\intercal \mathbf{1} \tag{77}$$
$$u_0 = 1 - \mathbf{1}^\intercal \mathbf{P} \operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}, \tag{78}$$

and similarly for the adversary's variables, where $\mathbf{Q} = \mathbf{Q}_1$:

$$\mathbf{Q}_0 = \mathbf{1}\mathbf{1}^\intercal \mathbf{Q} \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q} \tag{79}$$
$$\mathbf{s} = \operatorname{diag}(\boldsymbol{\kappa})\mathbf{Q}^\intercal \mathbf{1} \tag{80}$$
$$v_0 = 1 - \mathbf{1}^\intercal \mathbf{Q} \operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}. \tag{81}$$

Using this notation, we write Eq. (74) as:

$$\begin{aligned}
\min_{\mathbf{Q}\in\Delta} \max_{\mathbf{P}\in\Delta}\ & \langle \mathbf{M}_1, \mathbf{P}^\intercal \mathbf{Q}\rangle + \langle \mathbf{M}_2, \mathbf{P}^\intercal(\mathbf{1}\mathbf{1}^\intercal \mathbf{Q} \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q})\rangle + \langle \mathbf{M}_3, (\mathbf{1}\mathbf{1}^\intercal \mathbf{P} \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{P})^\intercal \mathbf{Q}\rangle \\
&+ \langle \mathbf{M}_4, (\mathbf{1}\mathbf{1}^\intercal \mathbf{P} \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{P})^\intercal (\mathbf{1}\mathbf{1}^\intercal \mathbf{Q} \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q})\rangle + \langle \mathbf{M}_5, \operatorname{diag}(\boldsymbol{\kappa})\mathbf{P}^\intercal \mathbf{1}\mathbf{1}^\intercal \mathbf{Q} \operatorname{diag}(\boldsymbol{\kappa})\rangle \\
&+ (1 - \mathbf{1}^\intercal \mathbf{P} \operatorname{diag}(\boldsymbol{\kappa})\mathbf{1})(1 - \mathbf{1}^\intercal \mathbf{Q} \operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}) - \langle \mathbf{Q}, \boldsymbol{\Omega}\rangle
\end{aligned} \tag{82}$$

The constraint set Δ for $\mathbf{P}$ is:

$$\begin{aligned}
&\mathbf{P} \geq 0 \\
&\mathbf{1}^\intercal \mathbf{P} \operatorname{diag}(\boldsymbol{\kappa})\mathbf{1} \leq 1 \\
&\mathbf{P} \leq \mathbf{1}\mathbf{1}^\intercal \mathbf{P} \operatorname{diag}(\boldsymbol{\kappa}),
\end{aligned}$$

and similarly for $\mathbf{Q}$:

$$\begin{aligned}
&\mathbf{Q} \geq 0 \\
&\mathbf{1}^\intercal \mathbf{Q} \operatorname{diag}(\boldsymbol{\kappa})\mathbf{1} \leq 1 \\
&\mathbf{Q} \leq \mathbf{1}\mathbf{1}^\intercal \mathbf{Q} \operatorname{diag}(\boldsymbol{\kappa}),
\end{aligned}$$

where all of the inequalities are element-wise. This matrix inequalities for defining Δ is equivalent with the inequalities in Eq. (10).

By rearranging the variables, we write Eq. (82) as:

$$\begin{aligned}
\min_{\mathbf{Q}\in\Delta} \max_{\mathbf{P}\in\Delta}\ & \langle \mathbf{P}, \mathbf{Q}\mathbf{M}_1^\intercal\rangle + \langle \mathbf{P}, (\mathbf{1}\mathbf{1}^\intercal \mathbf{Q} \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q})\mathbf{M}_2^\intercal\rangle + \langle \mathbf{1}\mathbf{1}^\intercal \mathbf{P} \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{P}, \mathbf{Q}\mathbf{M}_3^\intercal\rangle \\
&+ \langle \mathbf{1}\mathbf{1}^\intercal \mathbf{P} \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{P}, (\mathbf{1}\mathbf{1}^\intercal \mathbf{Q} \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q})\mathbf{M}_4^\intercal\rangle + \langle \mathbf{P}, \mathbf{1}\mathbf{1}^\intercal \mathbf{Q} \operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_5^\intercal \operatorname{diag}(\boldsymbol{\kappa})\rangle \\
&+ \langle \mathbf{P}, \mathbf{1}\mathbf{1}^\intercal \mathbf{Q} \operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{1}^\intercal \operatorname{diag}(\boldsymbol{\kappa})\rangle - \langle \mathbf{P}, \mathbf{1}\mathbf{1}^\intercal \operatorname{diag}(\boldsymbol{\kappa})\rangle - \langle \mathbf{Q}, \mathbf{1}\mathbf{1}^\intercal \operatorname{diag}(\boldsymbol{\kappa})\rangle + 1 - \langle \mathbf{Q}, \boldsymbol{\Omega}\rangle
\end{aligned} \tag{83}$$

$$= \min_{\mathbf{Q}\in\Delta} \max_{\mathbf{P}\in\Delta} 1 - \langle \mathbf{Q}, \mathbf{1}\mathbf{1}^\intercal \operatorname{diag}(\boldsymbol{\kappa})\rangle - \langle \mathbf{Q}, \boldsymbol{\Omega}\rangle \tag{84}$$
$$+ \langle \mathbf{P}, \mathbf{Q}\mathbf{M}_1^\intercal + (\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q})\mathbf{M}_2^\intercal + \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_5^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa})\rangle - \mathbf{1}\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa})\rangle$$
$$+ \langle \mathbf{1}\mathbf{1}^\intercal\mathbf{P}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{P}, \mathbf{Q}\mathbf{M}_3^\intercal + (\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q})\mathbf{M}_4^\intercal\rangle$$

$$= \min_{\mathbf{Q}\in\Delta} \max_{\mathbf{P}\in\Delta} 1 - \langle \mathbf{Q}, \mathbf{1}\mathbf{1}^\intercal \operatorname{diag}(\boldsymbol{\kappa})\rangle - \langle \mathbf{Q}, \boldsymbol{\Omega}\rangle \tag{85}$$
$$+ \langle \mathbf{P}, \mathbf{Q}\mathbf{M}_1^\intercal + (\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q})\mathbf{M}_2^\intercal + \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_5^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa})\rangle - \mathbf{1}\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa})\rangle$$
$$+ \langle \mathbf{P}, \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\mathbf{M}_3^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + \mathbf{1}\mathbf{1}^\intercal\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_4^\intercal\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\mathbf{M}_4^\intercal\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q}\mathbf{M}_3^\intercal - \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_4^\intercal + \mathbf{Q}\mathbf{M}_4^\intercal\rangle$$

Given a fixed $\mathbf{Q}$ maximizing $\mathbf{P} \in \Delta$ over a linear objective reduces to finding the column k that has the maximum sum of k largest elements in the column, with the additional restriction that it has to be greater than zero. We then simplify the formulation above as:

$$\min_{\mathbf{Q}\in\Delta} f(\mathbf{A}\mathbf{Q}\mathbf{B} + \mathbf{Q}\mathbf{C} + \mathbf{D}) + \langle \mathbf{Q}, \mathbf{E}\rangle + c \tag{86}$$

where:

$$f(\mathbf{X}) = \max(0, \max_k \text{ sum-k-largest}(\mathbf{X}_{(:,k)})) \tag{87}$$
$$\mathbf{A} = \mathbf{1}\mathbf{1}^\intercal \tag{88}$$
$$\mathbf{B} = \operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_2^\intercal + \operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_5^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + \operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + \mathbf{M}_3^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + n\operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_4^\intercal\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{M}_4^\intercal\operatorname{diag}(\boldsymbol{\kappa}) - \operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_4^\intercal \tag{89}$$
$$\mathbf{C} = \mathbf{M}_1^\intercal - \mathbf{M}_2^\intercal - \mathbf{M}_3^\intercal + \mathbf{M}_4^\intercal \tag{90}$$
$$\mathbf{D} = -\mathbf{1}\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa}) \tag{91}$$
$$\mathbf{E} = -\mathbf{1}\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa}) - \boldsymbol{\Omega} \tag{92}$$
$$c = 1 \tag{93}$$

D.1.2 ADMM Formulation

We perform an alternating direction method of multipliers (ADMM) optimization to optimize Eq. (86). We split the optimization into three variables: $\mathbf{Q}, \mathbf{X}$, and $\mathbf{Z}$.

$$\min_{\mathbf{Q},\mathbf{X},\mathbf{Z}} f(\mathbf{Z}) + \langle \mathbf{Q}, \mathbf{E}\rangle + \mathbf{I}_\Delta(\mathbf{Q}) + c \tag{94}$$
$$\text{s.t. } \mathbf{Z} = \mathbf{A}\mathbf{X}\mathbf{B} + \mathbf{X}\mathbf{C} + \mathbf{D}$$
$$\mathbf{Q} = \mathbf{X},$$

where $\mathbf{I}_\Delta(\mathbf{Q})$ returns 0 if $\mathbf{Q} \in \Delta$ or ∞ otherwise.

The augmented Lagrangian (scaled version) for this optimization is:

$$\mathcal{L}(\mathbf{Q}, \mathbf{X}, \mathbf{Z}, \mathbf{U}, \mathbf{W}) =$$
$$f(\mathbf{Z}) + \langle \mathbf{Q}, \mathbf{E}\rangle + \mathbf{I}_\Delta(\mathbf{Q}) + c + \frac{\rho}{2}\|\mathbf{A}\mathbf{X}\mathbf{B} + \mathbf{X}\mathbf{C} + \mathbf{D} - \mathbf{Z} + \mathbf{U}\|_F^2 + \frac{\rho}{2}\|\mathbf{X} - \mathbf{Q} + \mathbf{W}\|_F^2, \tag{95}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, ρ is the ADMM penalty parameter, whereas $\mathbf{U}$ and $\mathbf{W}$ are the dual variables for the constraint $\mathbf{Z} = \mathbf{A}\mathbf{X}\mathbf{B} + \mathbf{X}\mathbf{C} + \mathbf{D}$ and $\mathbf{Q} = \mathbf{X}$ respectively.

The ADMM updates for each variable are explained below:

1. Update for $\mathbf{Q}$: a projection operation

$$\mathbf{Q}^{(t+1)} = \operatorname*{argmin}_{\mathbf{Q}} \left\{ \langle \mathbf{Q}, \mathbf{E}\rangle + \mathbf{I}_\Delta(\mathbf{Q}) + \frac{\rho}{2}\|\mathbf{X}^{(t)} - \mathbf{Q} + \mathbf{W}^{(t)}\|_F^2 \right\} \tag{96}$$
$$= \operatorname*{argmin}_{\mathbf{Q}\in\Delta} \frac{1}{2}\|\tfrac{1}{\rho}(\rho(\mathbf{X}^{(t)} + \mathbf{W}^{(t)}) - \mathbf{E}) - \mathbf{Q}\|_F^2 \tag{97}$$
$$= \operatorname{Proj}_\Delta(\tfrac{1}{\rho}(\rho(\mathbf{X}^{(t)} + \mathbf{W}^{(t)}) - \mathbf{E})) \tag{98}$$

2. Update for $\mathbf{Z}$: a proximal operation.

$$\mathbf{Z}^{(t+1)} = \operatorname*{argmin}_{\mathbf{Z}} \left\{ f(\mathbf{Z}) + \frac{\rho}{2} \|\mathbf{A}\mathbf{X}^{(t)}\mathbf{B} + \mathbf{X}^{(t)}\mathbf{C} + \mathbf{D} - \mathbf{Z} + \mathbf{U}^{(t)}\|_F^2 \right\} \tag{99}$$

$$= \operatorname{prox}_{f,1/\rho}(\mathbf{A}\mathbf{X}^{(t)}\mathbf{B} + \mathbf{X}^{(t)}\mathbf{C} + \mathbf{D} + \mathbf{U}^{(t)}) \tag{100}$$

3. Update for $\mathbf{X}$: Sylvester equation

$$\mathbf{X}^{(t+1)} = \operatorname*{argmin}_{\mathbf{X}} \left\{ \frac{\rho}{2} \|\mathbf{A}\mathbf{X}\mathbf{B} + \mathbf{X}\mathbf{C} + \mathbf{D} - \mathbf{Z}^{(t+1)} + \mathbf{U}^{(t)}\|_F^2 + \frac{\rho}{2} \|\mathbf{X} - \mathbf{Q}^{(t+1)} + \mathbf{W}^{(t)}\|_F^2 \right\} \tag{101}$$

$$= \operatorname*{argmin}_{\mathbf{X}} \left\{ \frac{1}{2} \|\mathbf{A}\mathbf{X}\mathbf{B} + \mathbf{X}\mathbf{C} + \mathbf{D} - \mathbf{Z}^{(t+1)} + \mathbf{U}^{(t)}\|_F^2 + \frac{1}{2} \|\mathbf{X} - \mathbf{Q}^{(t+1)} + \mathbf{W}^{(t)}\|_F^2 \right\} \tag{102}$$

We solve the minimization above by setting the gradient w.r.t. $\mathbf{X}$ to zero. Removing the superscript over iteration t, the gradient of the objective above w.r.t. $\mathbf{X}$ is:

$$\mathbf{A}^\intercal\mathbf{A}\mathbf{X}\mathbf{B}\mathbf{B}^\intercal + \mathbf{A}^\intercal\mathbf{X}\mathbf{C}\mathbf{B}^\intercal + \mathbf{A}\mathbf{X}\mathbf{B}\mathbf{C}^\intercal + \mathbf{X}\mathbf{C}\mathbf{C}^\intercal + \mathbf{A}^\intercal(\mathbf{D} - \mathbf{Z} + \mathbf{U})\mathbf{B}^\intercal + (\mathbf{D} - \mathbf{Z} + \mathbf{U})\mathbf{C}^\intercal + \mathbf{X} + \mathbf{W} - \mathbf{Q}. \tag{103}$$

Since $\mathbf{A} = \mathbf{1}\mathbf{1}^\intercal$, the gradient can be simplified as:

$$\mathbf{A}\mathbf{X}n\mathbf{B}\mathbf{B}^\intercal + \mathbf{A}\mathbf{X}\mathbf{C}\mathbf{B}^\intercal + \mathbf{A}\mathbf{X}\mathbf{B}\mathbf{C}^\intercal + \mathbf{X}\mathbf{C}\mathbf{C}^\intercal + \mathbf{A}(\mathbf{D} - \mathbf{Z} + \mathbf{U})\mathbf{B}^\intercal + (\mathbf{D} - \mathbf{Z} + \mathbf{U})\mathbf{C}^\intercal + \mathbf{X} + \mathbf{W} - \mathbf{Q} \tag{104}$$

$$= \mathbf{A}\mathbf{X}(n\mathbf{B}\mathbf{B}^\intercal + \mathbf{C}\mathbf{B}^\intercal + \mathbf{B}\mathbf{C}^\intercal) + \mathbf{X}(\mathbf{C}\mathbf{C}^\intercal + \mathbf{I}) + \mathbf{A}(\mathbf{D} - \mathbf{Z} + \mathbf{U})\mathbf{B}^\intercal + (\mathbf{D} - \mathbf{Z} + \mathbf{U})\mathbf{C}^\intercal + \mathbf{W} - \mathbf{Q}. \tag{105}$$

Let $\mathbf{F} = \mathbf{A}(\mathbf{D} - \mathbf{Z} + \mathbf{U})\mathbf{B}^\intercal + (\mathbf{D} - \mathbf{Z} + \mathbf{U})\mathbf{C}^\intercal + \mathbf{W} - \mathbf{Q}$. The optimal $\mathbf{X}$ can be found by solving a Sylvester equation below:

$$\mathbf{A}\mathbf{X}(n\mathbf{B}\mathbf{B}^\intercal + \mathbf{C}\mathbf{B}^\intercal + \mathbf{B}\mathbf{C}^\intercal) + \mathbf{X}(\mathbf{C}\mathbf{C}^\intercal + \mathbf{I}) + \mathbf{F} = 0 \tag{106}$$

$$\mathbf{A}\mathbf{X}(n\mathbf{B}\mathbf{B}^\intercal + \mathbf{C}\mathbf{B}^\intercal + \mathbf{B}\mathbf{C}^\intercal) + \mathbf{X}(\mathbf{C}\mathbf{C}^\intercal + \mathbf{I}) = -\mathbf{F} \tag{107}$$

$$\mathbf{A}\mathbf{X}(n\mathbf{B}\mathbf{B}^\intercal + \mathbf{C}\mathbf{B}^\intercal + \mathbf{B}\mathbf{C}^\intercal)(\mathbf{C}\mathbf{C}^\intercal + \mathbf{I})^{-1} + \mathbf{X} = -\mathbf{F}(\mathbf{C}\mathbf{C}^\intercal + \mathbf{I})^{-1}. \tag{108}$$

Note that a Sylvester equation is a matrix equation in the form of $AXB + X = C$ or $AX + XB = C$.

4. Update for $\mathbf{U}$:

$$\mathbf{U}^{(t+1)} = \mathbf{U}^{(t)} + \mathbf{A}\mathbf{X}^{(t)}\mathbf{B} + \mathbf{X}^{(t)}\mathbf{C} + \mathbf{D} - \mathbf{Z}^{(t+1)}. \tag{109}$$

5. Update for $\mathbf{W}$:

$$\mathbf{W}^{(t+1)} = \mathbf{W}^{(t)} + \mathbf{X}^{(t+1)} - \mathbf{Q}^{(t+1)}. \tag{110}$$

Please go to Section D.4, D.5, and D.6 for the detailed algorithms for the projection, proximal operator, and Sylvester equation solver.

D.2 ADMM Formulation for Metrics without Special Cases

For the metric that does not enforce any special cases, the optimization over $\mathbf{Q}$ is:

$$\min_{\mathbf{Q}\in\Delta} \max_{\mathbf{P}\in\Delta} \left[\sum_{k,l\in[0,n]} \sum_{j} \frac{1}{g_j(k,l)} \left\{ a_j[\mathbf{p}_k^1 \cdot \mathbf{q}_l^1] + b_j[\mathbf{p}_k^0 \cdot \mathbf{q}_l^0] + c_j[\mathbf{p}_k^1 \cdot \mathbf{q}_l^0] + d_j[\mathbf{p}_k^0 \cdot \mathbf{q}_l^1] + f_j(k,l) r_k s_l \right\} - \langle \mathbf{Q}^\intercal\mathbf{1}, \Psi^\intercal\theta \rangle \right]. \tag{111}$$

Since the summation index in the equation above is from 0 to n, whereas our variables $\mathbf{P}$ and $\mathbf{Q}$ represent the indices from 1 to n, we need to treat the summation over index 0 separately. Specifically, the matrix notation optimization is now:

$$\min_{\{\mathbf{Q}_1,\mathbf{Q}_0,\mathbf{s},v_0\}\in\Delta} \max_{\{\mathbf{P}_1,\mathbf{P}_0,\mathbf{r},u_0\}\in\Delta} \langle \mathbf{M}_1, \mathbf{P}_1^\intercal\mathbf{Q}_1 \rangle + \langle \mathbf{M}_2, \mathbf{P}_1^\intercal\mathbf{Q}_0 \rangle + \langle \mathbf{M}_3, \mathbf{P}_0^\intercal\mathbf{Q}_1 \rangle + \langle \mathbf{M}_4, \mathbf{P}_0^\intercal\mathbf{Q}_0 \rangle + \langle \mathbf{M}_5, \mathbf{r}\mathbf{s}^\intercal \rangle \tag{112}$$

$$+ m_{4[0,0]} u_0 v_0 + \langle \mathbf{m}_{4[0,:]}, u_0 \mathbf{1}^\intercal\mathbf{Q}_0 \rangle + \langle \mathbf{m}_{4[:,0]}, \mathbf{P}_0^\intercal\mathbf{1} v_0 \rangle$$

$$+ m_{5[0,0]} u_0 v_0 + \langle \mathbf{m}_{5[0,:]}, u_0 \mathbf{s}^\intercal \rangle + \langle \mathbf{m}_{5[:,0]}, \mathbf{r} v_0 \rangle - \langle \mathbf{Q}_1, \mathbf{\Omega} \rangle,$$

where:

$$m_{4[0,0]} = \sum_j \frac{d_j}{g_j(0,0)}, \quad \mathbf{m}_{4[0,l]} = \sum_j \frac{d_j}{g_j(0,l)}, \quad \mathbf{m}_{4[k,0]} = \sum_j \frac{d_j}{g_j(k,0)},$$
$$m_{5[0,0]} = \sum_j \frac{f_j(0,0)}{g_j(0,0)}, \quad \mathbf{m}_{5[0,l]} = \sum_j \frac{f_j(0,l)}{g_j(0,l)}, \quad \mathbf{m}_{5[k,0]} = \sum_j \frac{f_j(k,0)}{g_j(k,0)}.$$

Using the same technique as in Appendix D.1, we write the optimization over the matrix $\mathbf{P}$ and $\mathbf{Q}$ only, and regroup the variables as follows:

$$\begin{aligned}
&\min_{\mathbf{Q}\in\Delta}\max_{\mathbf{P}\in\Delta} \langle \mathbf{M}_1, \mathbf{P}^\intercal\mathbf{Q}\rangle + \langle \mathbf{M}_2, \mathbf{P}^\intercal(\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q})\rangle + \langle \mathbf{M}_3, (\mathbf{1}\mathbf{1}^\intercal\mathbf{P}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{P})^\intercal\mathbf{Q}\rangle \qquad (113)\\
&\quad + \langle \mathbf{M}_4, (\mathbf{1}\mathbf{1}^\intercal\mathbf{P}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{P})^\intercal(\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q})\rangle + \langle \mathbf{M}_5, \operatorname{diag}(\boldsymbol{\kappa})\mathbf{P}^\intercal\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\rangle\\
&\quad + (m_{4[0,0]} + m_{5[0,0]})(1 - \mathbf{1}^\intercal\mathbf{P}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1})(1 - \mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1})\\
&\quad + \langle \mathbf{m}_{4[0,:]}, (1 - \mathbf{1}^\intercal\mathbf{P}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1})\mathbf{1}^\intercal(\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q})\rangle + \langle \mathbf{m}_{4[:,0]}, (\mathbf{1}\mathbf{1}^\intercal\mathbf{P}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{P})^\intercal\mathbf{1}(1 - \mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1})\rangle\\
&\quad + \langle \mathbf{m}_{5[0,:]}, (1 - \mathbf{1}^\intercal\mathbf{P}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1})\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\rangle + \langle \mathbf{m}_{5[:,0]}, \operatorname{diag}(\boldsymbol{\kappa})\mathbf{P}^\intercal\mathbf{1}(1 - \mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1})\rangle - \langle \mathbf{Q}, \boldsymbol{\Omega}\rangle
\end{aligned}$$

$$\begin{aligned}
=\ &\min_{\mathbf{Q}\in\Delta}\max_{\mathbf{P}\in\Delta} m_{4[0,0]} + m_{5[0,0]} - \langle \mathbf{Q}, \mathbf{1}\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa})(m_{4[0,0]} + m_{5[0,0]})\rangle - \langle \mathbf{Q}, \boldsymbol{\Omega}\rangle \qquad (114)\\
&\quad + \left\langle \mathbf{Q}, n\mathbf{1}\mathbf{m}_{4[0,:]}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{1}\mathbf{m}_{4[0,:]} + \mathbf{1}\mathbf{m}_{5[0,:]}\operatorname{diag}(\boldsymbol{\kappa})\right\rangle\\
&\quad + \Big\langle \mathbf{P}, \Big\{ \mathbf{Q}\mathbf{M}_1^\intercal + (\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q})\mathbf{M}_2^\intercal + \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_5^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa})(m_{4[0,0]} + m_{5[0,0]})\rangle\\
&\qquad - \mathbf{1}\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa})(m_{4[0,0]} + m_{5[0,0]}) - n\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{m}_{4[0,:]}^\intercal\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\mathbf{m}_{4[0,:]}^\intercal\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa})\\
&\qquad - \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{m}_{5[0,:]}^\intercal\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + \mathbf{1}\mathbf{m}_{5[:,0]}^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{m}_{5[:,0]}^\intercal\operatorname{diag}(\boldsymbol{\kappa})\Big\}\Big\rangle\\
&\quad + \Big\langle \mathbf{1}\mathbf{1}^\intercal\mathbf{P}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{P}, \Big\{ \mathbf{Q}\mathbf{M}_3^\intercal + (\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q})\mathbf{M}_4^\intercal + \mathbf{1}\mathbf{m}_{4[:,0]}^\intercal - \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{m}_{4[:,0]}^\intercal\Big\}\Big\rangle
\end{aligned}$$

$$\begin{aligned}
=\ &\min_{\mathbf{Q}\in\Delta}\max_{\mathbf{P}\in\Delta} m_{4[0,0]} + m_{5[0,0]} + \left\langle \mathbf{Q}, \left\{ n\mathbf{1}\mathbf{m}_{4[0,:]}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{1}\mathbf{m}_{4[0,:]} + \mathbf{1}\mathbf{m}_{5[0,:]}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{1}\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa})(m_{4[0,0]} + m_{5[0,0]}) - \boldsymbol{\Omega}\right\}\right\rangle\\
&\quad + \Big\langle \mathbf{P}, \Big\{ \mathbf{Q}\mathbf{M}_1^\intercal + (\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q})\mathbf{M}_2^\intercal + \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_5^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa})(m_{4[0,0]} + m_{5[0,0]})\rangle\\
&\qquad - \mathbf{1}\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa})(m_{4[0,0]} + m_{5[0,0]}) - n\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{m}_{4[0,:]}^\intercal\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\mathbf{m}_{4[0,:]}^\intercal\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa})\\
&\qquad - \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{m}_{5[0,:]}^\intercal\mathbf{1}^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + \mathbf{1}\mathbf{m}_{5[:,0]}^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{m}_{5[:,0]}^\intercal\operatorname{diag}(\boldsymbol{\kappa})\Big\}\Big\rangle\\
&\quad + \Big\langle \mathbf{P}, \Big\{ \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\mathbf{M}_3^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + n\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_4^\intercal\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\mathbf{M}_4^\intercal\operatorname{diag}(\boldsymbol{\kappa}) + n\mathbf{1}\mathbf{m}_{4[:,0]}^\intercal\operatorname{diag}(\boldsymbol{\kappa})\\
&\qquad - n\mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{m}_{4[:,0]}^\intercal\operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{Q}\mathbf{M}_3^\intercal - \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_4^\intercal + \mathbf{Q}\mathbf{M}_4^\intercal - \mathbf{1}\mathbf{m}_{4[:,0]}^\intercal + \mathbf{1}\mathbf{1}^\intercal\mathbf{Q}\operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{m}_{4[:,0]}^\intercal\Big\}\Big\rangle
\end{aligned} \qquad (115)$$

As in Appendix D.1, the equation above can be simplified as:

$$\min_{\mathbf{Q}\in\Delta} f(\mathbf{A}\mathbf{Q}\mathbf{B} + \mathbf{Q}\mathbf{C} + \mathbf{D}) + \langle \mathbf{Q}, \mathbf{E}\rangle + c \qquad (116)$$

where:

$$f(\mathbf{X}) = \max(0, \max_k \text{ sum-k-largest}(\mathbf{X}_{(:,k)})) \tag{117}$$

$$\mathbf{A} = \mathbf{1}\mathbf{1}^\intercal \tag{118}$$

$$\begin{aligned}\mathbf{B} = {} & \operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_2^\intercal + \operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_5^\intercal \operatorname{diag}(\boldsymbol{\kappa}) + \operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{1}^\intercal \operatorname{diag}(\boldsymbol{\kappa})(m_{4[0,0]} + m_{5[0,0]}) \\ & - n \operatorname{diag}(\boldsymbol{\kappa})\mathbf{m}_{4[0,:]}^\intercal \mathbf{1}^\intercal \operatorname{diag}(\boldsymbol{\kappa}) + \mathbf{m}_{4[0,:]}^\intercal \mathbf{1}^\intercal \operatorname{diag}(\boldsymbol{\kappa}) - \operatorname{diag}(\boldsymbol{\kappa})\mathbf{m}_{5[0,:]}^\intercal \mathbf{1}^\intercal \operatorname{diag}(\boldsymbol{\kappa}) + \operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{m}_{5[:,0]}^\intercal \operatorname{diag}(\boldsymbol{\kappa}) \\ & + \mathbf{M}_3^\intercal \operatorname{diag}(\boldsymbol{\kappa}) + n \operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_4^\intercal \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{M}_4^\intercal \operatorname{diag}(\boldsymbol{\kappa}) - n \operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{m}_{4[:,0]}^\intercal \operatorname{diag}(\boldsymbol{\kappa}) - \operatorname{diag}(\boldsymbol{\kappa})\mathbf{M}_4^\intercal + \operatorname{diag}(\boldsymbol{\kappa})\mathbf{1}\mathbf{m}_{4[:,0]}^\intercal\end{aligned} \tag{119}$$

$$\mathbf{C} = \mathbf{M}_1^\intercal - \mathbf{M}_2^\intercal - \mathbf{M}_3^\intercal + \mathbf{M}_4^\intercal \tag{120}$$

$$\mathbf{D} = -\mathbf{1}\mathbf{1}^\intercal \operatorname{diag}(\boldsymbol{\kappa})(m_{4[0,0]} + m_{5[0,0]}) + \mathbf{1}\mathbf{m}_{5[:,0]}^\intercal \operatorname{diag}(\boldsymbol{\kappa}) + n\mathbf{1}\mathbf{m}_{4[:,0]}^\intercal \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{1}\mathbf{m}_{4[:,0]}^\intercal \tag{121}$$

$$\mathbf{E} = n\mathbf{1}\mathbf{m}_{4[0,:]} \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{1}\mathbf{m}_{4[0,:]} + \mathbf{1}\mathbf{m}_{5[0,:]} \operatorname{diag}(\boldsymbol{\kappa}) - \mathbf{1}\mathbf{1}^\intercal \operatorname{diag}(\boldsymbol{\kappa})(m_{4[0,0]} + m_{5[0,0]}) - \boldsymbol{\Omega} \tag{122}$$

$$c = m_{4[0,0]} + m_{5[0,0]} \tag{123}$$

Since the form of the objective above is similar to the one in Appendix D.1, we use the same ADMM technique to solve the optimization over $\mathbf{Q}$. Note that only the constant variables that are defined by the form of the metric ($\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}$, and c) are modified from Eq. (86). All the ADMM updates remain the same.

D.3 ADMM Formulation for Metrics with Special Case for True Negative

For the metrics that enforce special cases for true negative only (e.g., specificity) or special cases for both true negative and true positive (e.g., the MCC and Kappa score), we use the optimization schemes for the metrics that do not enforce special cases for true negative, with a little modification. Specifically, we modify the coefficient matrix $\mathbf{M}_1$ and $\mathbf{M}_5$ by setting the values in the n-th row and the n-th column to be zero, except for the (n,n)-th cell where we set it to one. Therefore, for the metrics that enforce special cases for both true positive and true negative, we have:

$$\begin{aligned}\min_{\{\mathbf{Q}_1, \mathbf{Q}_0, \mathbf{s}, v_0\} \in \Delta} \max_{\{\mathbf{P}_1, \mathbf{P}_0, \mathbf{r}, u_0\} \in \Delta} \; & \langle \mathbf{M}_1^\diamond, \mathbf{P}_1^\intercal \mathbf{Q}_1 \rangle + \langle \mathbf{M}_2, \mathbf{P}_1^\intercal \mathbf{Q}_0 \rangle + \langle \mathbf{M}_3, \mathbf{P}_0^\intercal \mathbf{Q}_1 \rangle \\ & + \langle \mathbf{M}_4, \mathbf{P}_0^\intercal \mathbf{Q}_0 \rangle + \langle \mathbf{M}_5^\diamond, \mathbf{r}\mathbf{s}^\intercal \rangle + u_0 v_0 - \langle \mathbf{Q}_1, \Psi \rangle,\end{aligned} \tag{124}$$

whereas for the metrics that enforce special cases for true negative only we have:

$$\begin{aligned}\min_{\{\mathbf{Q}_1, \mathbf{Q}_0, \mathbf{s}, v_0\} \in \Delta} \max_{\{\mathbf{P}_1, \mathbf{P}_0, \mathbf{r}, u_0\} \in \Delta} \; & \langle \mathbf{M}_1^\diamond, \mathbf{P}_1^\intercal \mathbf{Q}_1 \rangle + \langle \mathbf{M}_2, \mathbf{P}_1^\intercal \mathbf{Q}_0 \rangle + \langle \mathbf{M}_3, \mathbf{P}_0^\intercal \mathbf{Q}_1 \rangle + \langle \mathbf{M}_4, \mathbf{P}_0^\intercal \mathbf{Q}_0 \rangle + \langle \mathbf{M}_5^\diamond, \mathbf{r}\mathbf{s}^\intercal \rangle \\ & + m_{4[0,0]} u_0 v_0 + \langle \mathbf{m}_{4[0,:]}, u_0 \mathbf{1}^\intercal \mathbf{Q}_0 \rangle + \langle \mathbf{m}_{4[:,0]}, \mathbf{P}_0^\intercal \mathbf{1} v_0 \rangle \\ & + m_{5[0,0]} u_0 v_0 + \langle \mathbf{m}_{5[0,:]}, u_0 \mathbf{s}^\intercal \rangle + \langle \mathbf{m}_{5[:,0]}, \mathbf{r} v_0 \rangle - \langle \mathbf{Q}_1, \Psi \rangle,\end{aligned} \tag{125}$$

where:

$$\mathbf{M}_{i,j}^\diamond = \begin{cases} 1 & \text{if } i = j = n \\ 0 & \text{if } (i = n \wedge j \neq n) \vee (i \neq n \wedge j = n) \\ \mathbf{M}_{i,j} & \text{otherwise.} \end{cases} \tag{126}$$

All other ADMM optimization techniques remain the same.

D.4 Projection onto the Valid Marginal Probability Set

In the ADMM updates for $\mathbf{Q}$ (Eq. (98)), we need to perform a projection onto the set of valid marginal distributions Δ. In this subsection, we will derive an algorithm to efficiently perform the projection.

Given a matrix $\mathbf{A}$ that is not necessary in the set Δ, we want to find $\mathbf{P} \in \Delta$ that minimizes the Euclidean distance between $\mathbf{A}$ and $\mathbf{P} \in \Delta$. Specifically, we need to solve:

$$\min_{\mathbf{P} \in \Delta} \tfrac{1}{2} \|\mathbf{P} - \mathbf{A}\|_F^2. \tag{127}$$

In our vector notation (see. Appendix A.2), this is equal to:

$$\begin{aligned} &\min_{\{\mathbf{p}_k\}} \tfrac{1}{2}\sum_k \|\mathbf{p}_k - \mathbf{a}_k\|_2^2 \\ &\text{subject to: } p_{i,k} \geq 0, \quad \forall i,k \in [1,n] \\ &\qquad p_{i,k} \leq \tfrac{1}{k}\textstyle\sum_j p_{j,k}, \quad \forall i,k \in [1,n] \\ &\qquad \textstyle\sum_k \tfrac{1}{k}\sum_i p_{i,k} \leq 1, \end{aligned} \tag{128}$$

where $\mathbf{p}_k$ and $\mathbf{a}_k$ are the k-th column of the $\mathbf{P}$ and $\mathbf{A}$ respectively.

The constraints above can be written as:

$$\begin{aligned} &\min_{\{\mathbf{p}_k \in \mathbb{C}_k\}} \tfrac{1}{2}\sum_k \|\mathbf{p}_k - \mathbf{a}_k\|_2^2, \text{ s.t. } \textstyle\sum_k \frac{\mathbf{p}_k^\intercal \mathbf{1}}{k} \leq 1 \\ &\text{where: } \mathbb{C}_k = \{\mathbf{p}_k \mid \mathbf{p}_k \in [0, r_k]^n;\ r_k \geq 0;\ r_k = \tfrac{\mathbf{p}_k^\intercal \mathbf{1}}{k}\}. \end{aligned} \tag{129}$$

Using the Lagrange multiplier technique, we write the dual optimization as:

$$\max_{\eta \geq 0} \min_{\{\mathbf{p}_k \in \mathbb{C}_k\}} \tfrac{1}{2}\sum_k \|\mathbf{p}_k - \mathbf{a}_k\|_2^2, + \eta\left(\textstyle\sum_k \frac{\mathbf{p}_k^\intercal \mathbf{1}}{k} - 1\right) \tag{130}$$

$$= \max_{\eta \geq 0} -\eta + \sum_k \min_{\mathbf{p}_k \in \mathbb{C}_k} \left\{\tfrac{1}{2}\|\mathbf{p}_k - \mathbf{a}_k\|_2^2 + \tfrac{\eta}{k}\mathbf{p}_k^\intercal \mathbf{1}\right\} \tag{131}$$

Given η, the inner minimization is now decomposable into each individual $\mathbf{p}_k$. For convenience, we drop the subscript k in the next analysis, i.e.,

$$\begin{aligned} &\min_{\mathbf{p} \in \mathbb{C}} \left\{\tfrac{1}{2}\|\mathbf{p} - \mathbf{a}\|_2^2 + \tfrac{\eta}{k}\mathbf{p}^\intercal \mathbf{1}\right\}. \\ &\text{where: } \mathbb{C} = \{\mathbf{p} \mid \mathbf{p} \in [0, r]^n;\ r \geq 0;\ r = \tfrac{\mathbf{p}^\intercal \mathbf{1}}{k}\}. \end{aligned} \tag{132}$$

This minimization problem admits a search-based analytical solution. We start with the $\bar{\mathbf{p}} = \mathbf{a} - \frac{\eta}{k}$, which is the minimizer of the objective without the constraint as the proposed solution, and start with $r = \frac{\bar{\mathbf{p}}^\intercal}{k}$. If all of p_i lies in $[0, r]$, we accept $\bar{\mathbf{p}}$ as the solution, otherwise, we iteratively reduce the value of the highest probability values in $\bar{\mathbf{p}}$, which automatically reduce the value of $r = \frac{\bar{\mathbf{p}}^\intercal}{k}$, and simultaneously setting negative values in $\bar{\mathbf{p}}$ as zero. This requires sorting the values in $\bar{\mathbf{p}}$ in a decreasing order.

Given we have the solution of Eq. (132) for each column, we calculate the objective and gradient of Eq. (132) with respect to η. Since it is just a one-dimensional optimization, we efficiently solve it with a gradient-based optimization with box constraint of $\eta \geq 0$. Note that the objective is concave with respect to η.

D.5 Proximal Operator for the ADMM Updates

In the ADMM updates for $\mathbf{Z}$ (Eq. (100)), we need to perform a proximal operator for the function $f(\mathbf{X})$, i.e.:

$$f(\mathbf{X}) = \max(0, \max_k \text{ sum-k-largest}(\mathbf{X}_{(:,k)})). \tag{133}$$

The proximal operator over f is:

$$\text{prox}_{f,1/\rho}(\mathbf{X}) = \operatorname*{argmin}_{\mathbf{Z}} \left\{f(\mathbf{Z}) + \frac{\rho}{2}\|\mathbf{X} - \mathbf{Z}\|_F^2\right\} \tag{134}$$

Note that $f(\mathbf{Z})$ can be expanded as:

$$f(\mathbf{Z}) = \max_{\mathbf{P} \in \Delta} \langle \mathbf{P}, \mathbf{Z}\rangle = \min_{\mathbf{P} \in \Delta} \langle \mathbf{P}, -\mathbf{Z}\rangle = \min_{\mathbf{P}} \left(\mathbf{I}_\Delta(\mathbf{P}) - \langle \mathbf{P}, \mathbf{Z}\rangle\right) = \sup_{\mathbf{P}} \left(\langle \mathbf{P}, \mathbf{Z}\rangle - \mathbf{I}_\Delta(\mathbf{P})\right) = \mathbf{I}_\Delta^*(\mathbf{Z}), \tag{135}$$

where $\mathbf{I}_\Delta^*(\mathbf{Z})$ denotes the conjugate function of $\mathbf{I}_\Delta(\mathbf{Z})$.

Based on Moreau Decomposition (Moreau, 1962), we know that:

$$\begin{aligned}
\text{prox}_f(\mathbf{X}) &= \mathbf{X} - \text{prox}_{\mathbf{I}_\Delta}(\mathbf{X}) && (136)\\
&= \mathbf{X} - \underset{\mathbf{Z}}{\text{argmin}} \left\{ \mathbf{I}_\Delta(\mathbf{Z}) + \tfrac{1}{2}\|\mathbf{X} - \mathbf{Z}\|_F^2 \right\} && (137)\\
&= \mathbf{X} - \underset{\mathbf{Z}\in\Delta}{\text{argmin}} \tfrac{1}{2}\|\mathbf{X} - \mathbf{Z}\|_F^2 && (138)\\
&= \mathbf{X} - \text{Proj}_\Delta(\mathbf{X}) && (139)
\end{aligned}$$

Therefore, we can compute $\text{prox}_{f,1/\rho}(\mathbf{X})$ as:

$$\begin{aligned}
\text{prox}_{f,1/\rho}(\mathbf{X}) &= \mathbf{X} - \tfrac{1}{\rho}\,\text{prox}_{\rho f^*}(\rho\mathbf{X}) && (140)\\
&= \mathbf{X} - \tfrac{1}{\rho}\,\text{Proj}_\Delta(\rho\mathbf{X}) && (141)
\end{aligned}$$

D.6 Solving the Sylvester Equation in the ADMM update

In the ADMM updates for $\mathbf{X}$ (Eq. (108)), we need solve a Sylvester equation in the form of:

$$\mathbf{A}\mathbf{X}(n\mathbf{B}\mathbf{B}^\intercal + \mathbf{C}\mathbf{B}^\intercal + \mathbf{B}\mathbf{C}^\intercal)(\mathbf{C}\mathbf{C}^\intercal + \mathbf{I})^{-1} + \mathbf{X} = -\mathbf{F}(\mathbf{C}\mathbf{C}^\intercal + \mathbf{I})^{-1}. \quad (142)$$

Many linear algebra packages in most of program languages have the capability to solve a Sylvester equation. However, since our formulation contains matrices with special property, we develop a faster customized solver that utilizes the eigen-decomposition technique and exploits the fact that $\mathbf{A}$, $(n\mathbf{B}\mathbf{B}^\intercal + \mathbf{C}\mathbf{B}^\intercal + \mathbf{B}\mathbf{C}^\intercal)$, and $(\mathbf{C}\mathbf{C}^\intercal + \mathbf{I})^{-1}$ are symmetric.

First, let us simplify the equation as:

$$\mathbf{A}\mathbf{X}\mathbb{B} + \mathbf{X} = \mathbb{F}, \quad (143)$$

where $\mathbb{B} = (n\mathbf{B}\mathbf{B}^\intercal + \mathbf{C}\mathbf{B}^\intercal + \mathbf{B}\mathbf{C}^\intercal)(\mathbf{C}\mathbf{C}^\intercal + \mathbf{I})^{-1}$ and $\mathbb{F} = -\mathbf{F}(\mathbf{C}\mathbf{C}^\intercal + \mathbf{I})^{-1}$. We perform eigen-decomposition on matrix $\mathbf{A}$ and $\mathbb{B}$, i.e.:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{U}^{-1}, \quad (144)$$

where $\mathbf{U}$ is a matrix whose i-th column is the eigenvector $\mathbf{u}_i$ of $\mathbf{A}$, and $\mathbf{S}$ is a diagonal matrix whose diagonal elements are the corresponding eigenvalues, $\mathbf{S}_{ii} = \lambda_i$. Similarly, we also have:

$$\mathbb{B} = \mathbf{V}\mathbf{T}\mathbf{V}^{-1}, \quad (145)$$

where $\mathbf{V}$ is a matrix whose i-th column is the eigenvector of $\mathbb{B}$, and $\mathbf{T}$ is a diagonal matrix whose diagonal elements are the corresponding eigenvalues of $\mathbb{B}$.

To make sure that we can apply the technique, we check the eigendecomposability of $\mathbf{A}$ and $\mathbb{B}$. Since $\mathbf{A}$ is symmetric, it is surely eigendecomposable. The matrix $\mathbb{B}$ may not be symmetric. However, both $\bar{\mathbf{B}} = (n\mathbf{B}\mathbf{B}^\intercal + \mathbf{C}\mathbf{B}^\intercal + \mathbf{B}\mathbf{C}^\intercal)$ and $\bar{\mathbf{C}} = (\mathbf{C}\mathbf{C}^\intercal + \mathbf{I})^{-1}$ are symmetric. Based on matrix similarity property, since $\mathbb{B} = \bar{\mathbf{B}}\bar{\mathbf{C}}$, the eigenvalues of $\mathbb{B}$ are the same as the eigenvalues of $\bar{\mathbf{C}}^{\frac{1}{2}}\bar{\mathbf{B}}\bar{\mathbf{C}}\bar{\mathbf{C}}^{-\frac{1}{2}} = \bar{\mathbf{C}}^{\frac{1}{2}}\bar{\mathbf{B}}\bar{\mathbf{C}}^{\frac{1}{2}}$, which is symmetric. Therefore, $\mathbb{B}$ is also eigendecomposable.

Applying the eigendecomposition technique, we have:

$$\begin{aligned}
\mathbf{A}\mathbf{X}\mathbb{B} + \mathbf{X} &= \mathbb{F} && (146)\\
\mathbf{U}\mathbf{S}\mathbf{U}^{-1}\mathbf{X}\mathbf{V}\mathbf{T}\mathbf{V}^{-1} + \mathbf{X} &= \mathbb{F} && (147)\\
\mathbf{U}\mathbf{S}(\mathbf{U}^{-1}\mathbf{X}\mathbf{V})\mathbf{T}\mathbf{V}^{-1} + \mathbf{X} &= \mathbb{F}. && (148)
\end{aligned}$$

Denote $\mathbf{X}^* = \mathbf{U}^{-1}\mathbf{X}\mathbf{V}$. We then have:

$$\begin{aligned}
\mathbf{U}\mathbf{S}\mathbf{X}^*\mathbf{T}\mathbf{V}^{-1} + \mathbf{X} &= \mathbb{F} && (149)\\
\mathbf{U}^{-1}\mathbf{U}\mathbf{S}\mathbf{X}^*\mathbf{T}\mathbf{V}^{-1}\mathbf{V} + \mathbf{U}^{-1}\mathbf{X}\mathbf{V} &= \mathbf{U}^{-1}\mathbb{F}\mathbf{V} && (150)\\
\mathbf{S}\mathbf{X}^*\mathbf{T} + \mathbf{X}^* &= \mathbf{U}^{-1}\mathbb{F}\mathbf{V} && (151)
\end{aligned}$$

Let $\mathbf{G} = \mathbf{U}^{-1}\mathbb{F}\mathbf{V}$. Since both $\mathbf{S}$ and $\mathbf{T}$ are diagonal matrices, we can solve for $\mathbf{X}^{\diamond}$ easily by solving element-wise equations, i.e.:

$$\mathbf{X}^{\diamond}_{i,j}(\mathbf{S}_{i,i}\mathbf{T}_{j,j} + 1) = \mathbf{G}_{i,j} \tag{152}$$

$$\mathbf{X}^{\diamond}_{i,j} = \frac{\mathbf{G}_{i,j}}{\mathbf{S}_{i,i}\mathbf{T}_{j,j} + 1}. \tag{153}$$

We can then easily recover $\mathbf{X}$ from $\mathbf{X}^{\diamond}$ by computing:

$$\mathbf{X} = \mathbf{U}\mathbf{X}^{\diamond}\mathbf{V}^{-1}. \tag{154}$$

When applying the decomposition technique above to the ADMM optimization, only the matrix $\mathbf{F}$ changes in each iteration. All other matrices are fixed based on the form of the optimized performance metric. Therefore, we only perform the eigendecomposition once and store most of the required variables for the computation. This left us with just a few matrix multiplication operations that need to be computed for each ADMM iteration.

D.7 Runtime Analysis

For a batch of m samples, all of the matrix variables in the ADMM formulations are $m \times m$ matrices. We run the ADMM algorithm for solving the inner optimization over $\mathbf{Q}$ in a fixed number of iterations (i.e., 100 iterations). In each iteration, we need to perform updates over the primal variables $\mathbf{Q}$, $\mathbf{Z}$, and $\mathbf{X}$. In updating $\mathbf{Q}$, we perform a projection algorithm to the set Δ. The runtime of the projection consists of sorting m-columns of m-items which costs $m^2 \log m$ in total. The iterative algorithm for finding the best $\mathbf{p}_k$ requires scanning the list, which costs $O(m)$ for each column, or $O(m^2)$ in total. The one-dimensional optimization for finding the optimal η converges very quickly. We cap the number of iterations of finding η to be at most 20 iterations. Hence, the total runtime of the projection algorithm is $O(m^2 \log m)$. The algorithm for computing the prox function in $\mathbf{Z}$ updates costs the same as the projection algorithm. For solving the Sylvester equation, we need to perform eigendecomposition once, which costs $O(m^3)$. For every ADMM iterations, we only need to perform a few matrix multiplication operations, which costs $O(m^{2.5})$. Therefore, the total runtime complexity for solving the inner optimization over $\mathbf{Q}$ using our ADMM algorithm is $O(m^3)$.