Supplementary Material

A. Proofs

In this appendix, we prove all theorems.

A.1. Proof of Theorem 1

On one hand, $\forall j \in [1, \dots, m]$ we have

$$\begin{aligned} \bar{\eta}_j(\boldsymbol{x}) &= \frac{p(\boldsymbol{x}, \bar{y} = j)}{\bar{p}(\boldsymbol{x})} \\ &= \frac{p(\boldsymbol{x} \mid \bar{y} = j) \cdot p(\bar{y} = j)}{\bar{p}(\boldsymbol{x})} \\ &= \frac{\rho_j \cdot [\pi_j \cdot p_p(\boldsymbol{x}) + (1 - \pi_j) \cdot p_n(\boldsymbol{x})]}{\sum_{j=1}^m \rho_j \cdot [\pi_j \cdot p_p(\boldsymbol{x}) + (1 - \pi_j) \cdot p_n(\boldsymbol{x})]}. \end{aligned} \tag{16}$$

In the third equality, we substitute $p(\boldsymbol{x} \mid \bar{y} = j)$ with p_{tr} that is defined in (3). On the other hand, by Bayes' rule we have

$$p_p(\boldsymbol{x}) = p(\boldsymbol{x} \mid y = +1) = \frac{p(y = +1 \mid \boldsymbol{x}) \cdot p(\boldsymbol{x})}{p(y = +1)} = \frac{\eta(\boldsymbol{x}) \cdot p(\boldsymbol{x})}{\pi_{\mathcal{D}}}, \tag{17}$$

$$p_n(\boldsymbol{x}) = p(\boldsymbol{x} \mid y = -1) = \frac{p(y = -1 \mid \boldsymbol{x}) \cdot p(\boldsymbol{x})}{p(y = -1)} = \frac{(1 - \eta(\boldsymbol{x})) \cdot p(\boldsymbol{x})}{1 - \pi_{\mathcal{D}}}. \tag{18}$$

Then, we plug (17) and (18) into (16) and obtain

$$\begin{aligned} \bar{\eta}_j(\boldsymbol{x}) &= \frac{\rho_j \cdot [\pi_j \eta(\boldsymbol{x}) \cdot (1 - \pi_{\mathcal{D}}) + (1 - \pi_j) \cdot (1 - \eta(\boldsymbol{x})) \cdot \pi_{\mathcal{D}}]}{\sum_{j=1}^m \rho_j \cdot [\pi_j \eta(\boldsymbol{x}) \cdot (1 - \pi_{\mathcal{D}}) + (1 - \pi_j) \cdot (1 - \eta(\boldsymbol{x})) \cdot \pi_{\mathcal{D}}]} \\ &= \frac{\rho_j \cdot (\pi_j - \pi_{\mathcal{D}}) \cdot \eta(\boldsymbol{x}) + \rho_j \cdot (1 - \pi_j) \cdot \pi_{\mathcal{D}}}{\sum_{j=1}^m \rho_j \cdot (\pi_j - \pi_{\mathcal{D}}) \cdot \eta(\boldsymbol{x}) + \sum_{j=1}^m \rho_j \cdot (1 - \pi_j) \cdot \pi_{\mathcal{D}}}. \end{aligned}$$

By setting the coefficients a_j, b_j, c, d accordingly we conclude the proof. □

A.2. Proof of Lemma 2

We proceed the proof by firstly showing that the denominator of each function $T_j(t)$, $j = 1, \dots, m$, is strictly greater than zero for all $t \in [0, 1]$, and then showing that $\boldsymbol{T}(t_1) = \boldsymbol{T}(t_2)$ if and only if $t_1 = t_2$.

For all $j = 1, \dots, m$, the denominators of $T_j(t)$ are the same, i.e., $c \cdot t + d$, where $c = \sum_{j=1}^m \rho_j(\pi_j - \pi_{\mathcal{D}})$ and $d = \sum_{j=1}^m \rho_j \pi_{\mathcal{D}}(1 - \pi_j)$. We know that d is positive because $\rho_j > 0$, $\pi_{\mathcal{D}} > 0$, and there exists $j \in 1, \dots, m$ such that $\pi_j < 1$. Given that $t \in [0, 1]$, we discuss the sign of c as follows:

1. if $c \geq 0$, the minimum value of $c \cdot t + d$ is $c \cdot 0 + d = d > 0$;
2. if $c < 0$, the minimum value of $c \cdot t + d$ is $c \cdot 1 + d = \sum_{j=1}^m \rho_j(\pi_j - \pi_{\mathcal{D}}) + \sum_{j=1}^m \rho_j \pi_{\mathcal{D}}(1 - \pi_j) = \sum_{j=1}^m \rho_j \pi_j (1 - \pi_{\mathcal{D}}) > 0$, where the last inequality is due to the existence of $j \in 1, \dots, m$ such that $\pi_j > 0$.

Hitherto, we have shown that the denominator $c \cdot t + d > 0$. Next, we prove the one-to-one mapping property by contradiction. Assume that there exist $t_1, t_2 \in [0, 1]$ such that $t_1 \neq t_2$ but $\boldsymbol{T}(t_1) = \boldsymbol{T}(t_2)$, which indicates that $T_j(t_1) = T_j(t_2), \forall j = 1, \dots, m$. For all j, we have

$$\begin{aligned} T_j(t_1) - T_j(t_2) &= \frac{a_j \cdot t_1 + b_j}{c \cdot t_1 + d} - \frac{a_j \cdot t_2 + b_j}{c \cdot t_2 + d} \\ &= \frac{(a_j \cdot t_1 + b_j)((c \cdot t_2 + d)) - (a_j \cdot t_2 + b_j)((c \cdot t_1 + d))}{(c \cdot t_1 + d)(c \cdot t_2 + d)} \\ &= \frac{(t_1 - t_2)(a_j \cdot d - b_j \cdot c)}{(c \cdot t_1 + d)(c \cdot t_2 + d)} \qquad (19) \\ &= 0, \end{aligned}$$

where $a_j = \rho_j \cdot (\pi_j - \pi_{\mathcal{D}})$ and $b_j = \rho_j \cdot (1 - \pi_j) \cdot \pi_{\mathcal{D}}$. As shown previously, the denominator of (19) is non-zero for all j. Next we show that there exists $j \in 1, \ldots, m$ such that $a_j \cdot d - b_j c \neq 0$. Since c and d are constants and irrelevant to i, we have

$$\begin{aligned} a_j \cdot d - b_j \cdot c &= (\rho_j \cdot (\pi_j - \pi_{\mathcal{D}})) \cdot d - (\rho_j \cdot (1 - \pi_j) \cdot \pi_{\mathcal{D}}) \cdot c \\ &= \rho_j \cdot (\pi_j \cdot d - \pi_{\mathcal{D}} \cdot d - c + \pi_j \cdot c) \\ &= \rho_j \cdot (\pi_j \cdot (c + d) - \pi_{\mathcal{D}} \cdot d - c). \end{aligned}$$

This equation equals to zero if and only if $\pi_j = \frac{c + \pi_{\mathcal{D}} \cdot d}{c + d}$. According to our assumption that at least two of the U sets are different, $\exists j' \in 1, \ldots, m$ such that $\pi_{j'} \neq \frac{c + \pi_{\mathcal{D}} \cdot d}{c + d}$. For such j', $T_{j'}(t_1) = T_{j'}(t_2)$ if and only if $t_1 = t_2$, which leads to a contradiction since $t_1 \neq t_2$. So we conclude the proof that $\boldsymbol{T}(t_1) = \boldsymbol{T}(t_2)$ if and only if $t_1 = t_2$. □

A.3. Proof of Lemma 3

We provide a proof of the cross-entropy loss and mean squared error, which are commonly used losses because of their numerical stability and good convergence rate (De Boer et al., 2005; Allen, 1971).

Cross-entropy loss Since the cross-entropy loss is non-negative by its definition, minimizing $R_{\text{surr}}(\boldsymbol{g})$ can be obtained by minimizing the conditional risk $\mathbb{E}_{p(\bar{y}|\boldsymbol{x})}[\ell(\boldsymbol{g}(\boldsymbol{x}), \bar{y}) \mid \boldsymbol{x}]$ for every $\boldsymbol{x} \in \mathcal{X}$. So we are now optimizing

$$\phi(\boldsymbol{g}) = -\sum_{j=1}^{m} p(\bar{y} = j \mid \boldsymbol{x}) \cdot \log(g_j(\boldsymbol{x})), \quad \text{s.t.} \sum_{j=1}^{m} g_j(\boldsymbol{x}) = 1.$$

By using the Lagrange multiplier method (Bertsekas, 1997), we have

$$\mathcal{L} = -\sum_{j=1}^{m} p(\bar{y} = j \mid \boldsymbol{x}) \cdot \log(g_j(\boldsymbol{x})) - \lambda \cdot (\sum_{j=1}^{m} g_j(\boldsymbol{x}) - 1).$$

The derivative of $\mathcal{L}$ with respect to $\boldsymbol{g}$ is

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{g}} = [-\frac{p(\bar{y} = 1 \mid \boldsymbol{x})}{g_1(\boldsymbol{x})} - \lambda, \cdots, -\frac{p(\bar{y} = m \mid \boldsymbol{x})}{g_m(\boldsymbol{x})} - \lambda]^\top.$$

By setting this derivative to 0 we obtain

$$g_j(\boldsymbol{x}) = -\frac{1}{\lambda} \cdot p(\bar{y} = j \mid \boldsymbol{x}), \quad \forall j = 1, \ldots, m \text{ and } \forall \boldsymbol{x} \in \mathcal{X}.$$

Since $\boldsymbol{g} \in \Delta^{m-1}$ is the m-dimensional simplex, we have $\sum_{j=1}^{m} g_j^\star(\boldsymbol{x}) = 1$ and $\sum_{j=1}^{m} p(\bar{y} = j \mid \boldsymbol{x}) = 1$. Then

$$\sum_{j=1}^{m} g_j^\star(\boldsymbol{x}) = -\frac{1}{\lambda} \cdot \sum_{j=1}^{m} p(\bar{y} = j \mid \boldsymbol{x}) = 1.$$

Therefore we obtain $\lambda = -1$ and $g_j^\star(\boldsymbol{x}) = p(\bar{y} = j \mid \boldsymbol{x}) = \bar{\eta}_j(\boldsymbol{x}), \forall j = 1, \ldots, m$ and $\forall \boldsymbol{x} \in \mathcal{X}$, which is equivalent to $\boldsymbol{g}^\star = \bar{\boldsymbol{\eta}}$. Note that when $m = 2$, the softmax is reduce to sigmoid function and the cross-entropy is reduced to logistic loss $\ell_{\log}(z) = \ln(1 + \exp(-z))$.

Mean squared error Similarly to the cross-entropy loss, we transform the risk minimization problem to the following constrained optimization problem

$$\phi(\boldsymbol{g}) = \sum_{j=1}^{m} (p(\bar{y} = j \mid \boldsymbol{x}) - g_j(\boldsymbol{x}))^2, \quad \text{s.t.} \sum_{j=1}^{m} g_j(\boldsymbol{x}) = 1.$$

By using the Lagrange multiplier method, we obtain

$$\mathcal{L} = \sum_{j=1}^{m} (p(\bar{y} = j \mid \boldsymbol{x}) - g_j(\boldsymbol{x}))^2 - \lambda \cdot (\sum_{j=1}^{m} g_j(\boldsymbol{x}) - 1).$$

The derivative of $\mathcal{L}$ with respect to $\boldsymbol{g}$ is

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{g}} = [2g_1(\boldsymbol{x}) - 2p(\bar{y}=1 \mid \boldsymbol{x}) - \lambda, \cdots, 2g_m(\boldsymbol{x}) - 2p(\bar{y}=m \mid \boldsymbol{x}) - \lambda]^\top.$$

By setting this derivative to 0 we obtain

$$g_j(\boldsymbol{x}) = p(\bar{y}=j \mid \boldsymbol{x}) + \frac{\lambda}{2}.$$

Since $\sum_{j=1}^m g_j^\star(\boldsymbol{x}) = 1$ and $\sum_{j=1}^m p(\bar{y}=j \mid \boldsymbol{x}) = 1$, we have

$$\begin{aligned}\sum_{j=1}^m g_j^\star(\boldsymbol{x}) &= \sum_{j=1}^m p(\bar{y}=j \mid \boldsymbol{x}) + \frac{\lambda \cdot m}{2}, \\ \frac{\lambda \cdot m}{2} &= 0.\end{aligned}$$

Since $m \geq 2$, we can obtain $\lambda = 0$. Consequently, $g_j^\star(\boldsymbol{x}) = p(\bar{y}=j \mid \boldsymbol{x}) = \bar{\eta}_j(\boldsymbol{x})$, which leads to $\boldsymbol{g}^\star = \bar{\boldsymbol{\eta}}$. We conclude the proof. □

A.4. Proof of Theorem 4

According to Lemma 3, when a cross-entropy loss or mean squared error is used for ℓ, the mapping $\boldsymbol{g}^\star(\boldsymbol{x}) = \bar{\boldsymbol{\eta}}(\boldsymbol{x})$ is the unique minimizer of $R_{\mathrm{surr}}(\boldsymbol{g}; \ell)$. Let $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{T}(f(\boldsymbol{x}))$, since $\boldsymbol{g}^\star \in \mathcal{G}$, $R_{\mathrm{surr}}(\boldsymbol{g}(\boldsymbol{x})) = \mathbb{E}_{(\boldsymbol{x},\bar{y})\sim\bar{\mathcal{D}}}[\ell(\boldsymbol{g}(\boldsymbol{x}), \bar{y})]$ achieves its minimum if and only if $\boldsymbol{g}(\boldsymbol{x}) = \bar{\boldsymbol{\eta}}(\boldsymbol{x}) = \boldsymbol{g}^\star(\boldsymbol{x})$. Combining this result with Theorem 1 and Lemma 2, we then obtain that $\boldsymbol{g}(\boldsymbol{x}) = \bar{\boldsymbol{\eta}}(\boldsymbol{x})$ if and only if $f(\boldsymbol{x}) = \eta(\boldsymbol{x})$. Since

$$\begin{aligned}R_{\mathrm{surr}}(f) &= \mathbb{E}_{(\boldsymbol{x},\bar{y})\sim\bar{\mathcal{D}}}[\ell(\boldsymbol{T}(f(\boldsymbol{x})), \bar{y})] \\ &= \mathbb{E}_{(\boldsymbol{x},\bar{y})\sim\bar{\mathcal{D}}}[\ell(\boldsymbol{g}(\boldsymbol{x}), \bar{y})] = R_{\mathrm{surr}}(\boldsymbol{g}),\end{aligned}$$

$f_{\mathrm{surr}}^\star$ is induced by $\boldsymbol{g}^\star = \operatorname{argmin}_{\boldsymbol{g}} R_{\mathrm{surr}}(\boldsymbol{g})$. So we have $f_{\mathrm{surr}}^\star(\boldsymbol{x}) = \operatorname{argmin}_f R_{\mathrm{surr}}(f) = \eta(\boldsymbol{x})$.

On the other hand, when ℓ_{b} is a cross-entropy loss, i.e., the logistic loss in the binary case, or mean squared error, the mapping $f^\star$ is the unique minimizer of $R(f; \ell_{\mathrm{b}})$. We skip the proof since it is similar to the proof of Lemma 3. So we obtain that $f^\star(\boldsymbol{x}) = \eta(\boldsymbol{x}) = f_{\mathrm{surr}}^\star(\boldsymbol{x})$, which concludes the proof. □

A.5. Proof of Lemma 5

$\forall j \in 1, \ldots, m$, by taking derivative of $T_j(t)$ with respect to t, we obtain

$$\left|\frac{\partial T_j}{\partial t}\right| = \frac{|a_j d - b_j c|}{(c \cdot t + d)^2}, \tag{20}$$

where

$$a_j = \rho_j(\pi_j - \pi_{\mathcal{D}}), \quad b_j = \rho_j \pi_{\mathcal{D}}(1 - \pi_j), \quad c = \sum_{j=1}^m \rho_j(\pi_j - \pi_{\mathcal{D}}), \quad d = \sum_{j=1}^m \rho_j \pi_{\mathcal{D}}(1 - \pi_j).$$

Since for all class priors we have $0 \leq \pi_j \leq 1$, $0 < \pi_D < 1$, $0 < \rho_j < 1$, $\sum_{j=1}^m \rho_j = 1$, and $\exists j, j' \in \{1, \ldots, m\}$ such that $j \neq j'$ and $\pi_j \neq \pi_{j'}$, obviously we can obtain

$$-1 \leq a_j \leq 1, \quad 0 \leq b_j \leq 1, \quad -1 \leq c \leq 1, \quad \text{and } 0 < d \leq 1.$$

Therefore, the numerator of (20) satisfies

$$|a_j d - b_j c| \leq 2. \tag{21}$$

On the other hand, since $d > 0$ and $0 \leq t \leq 1$, by substituting $t = 0$ and $t = 1$ respectively, we can obtain

$$\begin{aligned}c \cdot t + d \geq c + d = \sum_{j=1}^m \rho_j \pi_j (1 - \pi_{\mathcal{D}}) > 0, &\quad \text{if } c < 0; \\ c \cdot t + d \geq d > 0, &\quad \text{if } c \geq 0.\end{aligned}$$

Next we lower bound this term by $c \cdot t + d \geq \min(c+d, d) > 0$. As a result, the denominator of (20) satisfies

$$\begin{aligned}(c \cdot t + d)^2 &\geq (\min(c+d, d))^2 \\ &= \left(\min\left(\sum_{j=1}^{m} \rho_j \pi_j (1-\pi_{\mathcal{D}}), \sum_{j=1}^{m} \rho_j \pi_{\mathcal{D}} (1-\pi_j)\right)\right)^2 \\ &= \alpha^2.\end{aligned} \tag{22}$$

Then, by combing (21) and (22), we have

$$\left|\frac{\partial T_j}{\partial t}\right| \leq \frac{2}{\alpha^2}.$$

This bound illustrates that $T_j(f(\boldsymbol{x}))$ is Lipschitz-continuous with respect to $f(\boldsymbol{x})$ with a Lipschitz constant $2/\alpha^2$ and we complete the proof. □

A.6. Proof of Theorem 6

We first introduce the following lemmas which are useful to derive the estimation error bound.

Lemma 7 (Uniform deviation bound)**.** *Let $\boldsymbol{g} \in \mathcal{G}$, where $\mathcal{G} = \{\boldsymbol{x} \mapsto \boldsymbol{T}(f(\boldsymbol{x})) \mid f \in \mathcal{F}\}$ is a class of measurable functions, $\mathcal{X}_{\mathrm{tr}} = \{(\boldsymbol{x}_i, \bar{y}_i)\}_{i=1}^{n_{\mathrm{tr}}} \overset{\text{i.i.d.}}{\sim} \bar{\mathcal{D}}$ be a fixed sample of size n_{tr} i.i.d. drawn from $\bar{\mathcal{D}}$, and $\{\sigma_1, \ldots, \sigma_{n_{\mathrm{tr}}}\}$ be the Rademacher variables, i.e., independent uniform random variables taking values in $\{-1, 1\}$. Let $\mathfrak{R}_{n_{\mathrm{tr}}}(\ell \circ \mathcal{G})$ be the Rademacher complexity of $\ell \circ \mathcal{G}$ which is defined as*

$$\mathfrak{R}_{n_{\mathrm{tr}}}(\ell \circ \mathcal{G}) = \mathbb{E}\left[\sup_{\boldsymbol{g} \in \mathcal{G}} \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \sigma_i \ell(\boldsymbol{g}(\boldsymbol{x}_i), \bar{y}_i)\right].$$

Under the assumptions of Theorem 6, $\ell(\boldsymbol{g}(\boldsymbol{x}), \bar{y})$ is upper-bounded by M_ℓ. Then, for any $\delta > 0$, we have with probability at least $1-\delta$,

$$\sup_{\boldsymbol{g} \in \mathcal{G}} |\widehat{R}_{\mathrm{surr}}(\boldsymbol{g}) - R_{\mathrm{surr}}(\boldsymbol{g})| \leq 2\mathfrak{R}_{n_{\mathrm{tr}}}(\ell \circ \mathcal{G}) + M_\ell \sqrt{\frac{\ln(2/\delta)}{2n_{\mathrm{tr}}}}.$$

Proof. We consider the one-side uniform deviation $\sup_{\boldsymbol{g} \in \mathcal{G}} \widehat{R}_{\mathrm{surr}}(\boldsymbol{g}) - R_{\mathrm{surr}}(\boldsymbol{g})$. Suppose that a sample $(\boldsymbol{x}_i, \bar{y}_i)$ is replaced by another arbitrary sample $(\boldsymbol{x}_j, \bar{y}_j)$, the change of $\sup_{\boldsymbol{g} \in \mathcal{G}} \widehat{R}_{\mathrm{surr}}(\boldsymbol{g}) - R_{\mathrm{surr}}(\boldsymbol{g})$ is no more than M_ℓ/n_{tr}, since the loss $\ell(\cdot)$ is bounded by M_ℓ. By applying the McDiarmid's inequality (McDiarmid, 1989), for all $\epsilon' > 0$ we have

$$\Pr\{\sup_{\boldsymbol{g} \in \mathcal{G}} \widehat{R}_{\mathrm{surr}}(\boldsymbol{g}) - R_{\mathrm{surr}}(\boldsymbol{g}) - \mathbb{E}[\sup_{\boldsymbol{g} \in \mathcal{G}} \widehat{R}_{\mathrm{surr}}(\boldsymbol{g}) - R_{\mathrm{surr}}(\boldsymbol{g})] \geq \epsilon'\} \leq \exp\left(\frac{-2n_{\mathrm{tr}}\epsilon'^2}{M_\ell^2}\right).$$

Equivalently, for any $\delta > 0$, with probability at least $1 - \delta/2$,

$$\sup_{\boldsymbol{g} \in \mathcal{G}} \widehat{R}_{\mathrm{surr}}(\boldsymbol{g}) - R_{\mathrm{surr}}(\boldsymbol{g}) \leq \mathbb{E}\left[\sup_{\boldsymbol{g} \in \mathcal{G}} \widehat{R}_{\mathrm{surr}}(\boldsymbol{g}) - R_{\mathrm{surr}}(\boldsymbol{g})\right] + M_\ell \sqrt{\frac{\ln(2/\delta)}{2n_{\mathrm{tr}}}}.$$

By *symmetrization* (Vapnik, 1998), it is a routine work to show that

$$\mathbb{E}\left[\sup_{\boldsymbol{g} \in \mathcal{G}} \widehat{R}_{\mathrm{surr}}(\boldsymbol{g}) - R_{\mathrm{surr}}(\boldsymbol{g})\right] \leq 2\mathfrak{R}_{n_{\mathrm{tr}}}(\ell \circ \mathcal{G}).$$

The other side uniform deviation $\sup_{\boldsymbol{g} \in \mathcal{G}} R_{\mathrm{surr}}(\boldsymbol{g}) - \widehat{R}_{\mathrm{surr}}(\boldsymbol{g})$ can be bounded similarly. By combining the two sides' inequalities, we complete the proof. □

Lemma 8. *Let $f \in \mathcal{F}$, where $\mathcal{F} = \{f : \mathcal{X} \to \mathbb{R}\}$ is a class of measurable functions, $\{\boldsymbol{x}_i\}_{i=1}^{n_{\mathrm{tr}}} \overset{\text{i.i.d.}}{\sim} p_{\mathrm{tr}}(\boldsymbol{x})$ be a fixed sample of size n_{tr} i.i.d. drawn from the marginal density $p_{\mathrm{tr}}(\boldsymbol{x})$, and $\{\sigma_1, \dots, \sigma_{n_{\mathrm{tr}}}\}$ be the Rademacher variables. Let $\mathfrak{R}_{n_{\mathrm{tr}}}(\mathcal{F})$ be the Rademacher complexity of $\mathcal{F}$ which is defined as*

$$\mathfrak{R}_{n_{\mathrm{tr}}}(\mathcal{F}) = \mathbb{E}\left[\sup_{f \in \mathcal{F}} \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \sigma_i f(\boldsymbol{x}_i)\right].$$

Then we have

$$\mathfrak{R}_{n_{\mathrm{tr}}}(\ell \circ \mathcal{G}) \leq \frac{2\sqrt{2}m\mathcal{L}_\ell}{\alpha^2} \mathfrak{R}_{n_{\mathrm{tr}}}(\mathcal{F}).$$

Proof. In what follows, we upper-bound $\mathfrak{R}_{n_{\mathrm{tr}}}(\ell \circ \mathcal{G})$. Since $\ell(\boldsymbol{g}(\boldsymbol{x}), \bar{y})$ is $\mathcal{L}_\ell$-Lipschitz continuous w.r.t $\boldsymbol{g}$, according to the Rademacher vector contraction inequality (Maurer, 2016), we have

$$\begin{aligned}
\mathfrak{R}_{n_{\mathrm{tr}}}(\ell \circ \mathcal{G}) &= \mathbb{E}\left[\sup_{\boldsymbol{g} \in \mathcal{G}} \frac{1}{n_{\mathrm{tr}}} \sum_{i=1}^{n_{\mathrm{tr}}} \sigma_i \ell(\boldsymbol{g}(\boldsymbol{x}_i), \bar{y}_i)\right] \\
&\leq \frac{\sqrt{2}\mathcal{L}_\ell}{n_{\mathrm{tr}}} \cdot \mathbb{E}\left[\sup_{\boldsymbol{g} \in \mathcal{G}} \sum_{i=1}^{n_{\mathrm{tr}}} \sum_{j=1}^{m} \sigma_{ij} g_j(\boldsymbol{x}_i)\right] \\
&\leq \frac{\sqrt{2}\mathcal{L}_\ell}{n_{\mathrm{tr}}} \cdot \sum_{j=1}^{m} \mathbb{E}\left[\sup_{\boldsymbol{g} \in \mathcal{G}} \sum_{i=1}^{n_{\mathrm{tr}}} \sigma_{ij} g_j(\boldsymbol{x}_i)\right], \qquad (23)
\end{aligned}$$

where $g_j(\boldsymbol{x_i})$ is the j-th component of $\boldsymbol{g}(\boldsymbol{x_i})$, and σ_{ij} are an $n_{\mathrm{tr}} \times m$ matrix of independent Rademacher variables. As shown in Lemma 5, $g_j(\boldsymbol{x}) = T_j(f(\boldsymbol{x}))$ and $T_j(f)$ is Lipschitz continuous w.r.t f with a Lipschitz constant $2/\alpha^2$. Then we apply the Talagrand's contraction lemma (Shalev-Shwartz & Ben-David, 2014) and obtain

$$\begin{aligned}
\sum_{j=1}^{m} \mathbb{E}\left[\sup_{\boldsymbol{g} \in \mathcal{G}} \sum_{i=1}^{n_{\mathrm{tr}}} \sigma_{ij} g_j(\boldsymbol{x}_i)\right] &= \sum_{j=1}^{m} \mathbb{E}\left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n_{\mathrm{tr}}} \sigma_{ij} T_j(f(\boldsymbol{x_i}))\right] \\
&\leq \frac{2}{\alpha^2} \sum_{j=1}^{m} \mathbb{E}\left[\sup_{f \in \mathcal{F}} \sum_{i=1}^{n_{\mathrm{tr}}} \sigma_{ij} f(\boldsymbol{x_i})\right] \\
&= \frac{2mn_{\mathrm{tr}}}{\alpha^2} \mathfrak{R}_{n_{\mathrm{tr}}}(\mathcal{F}).
\end{aligned}$$

By substituting it into (23), we complete the proof. □

Based on Lemma 7 and Lemma 8, the estimation error bound is proven through

$$\begin{aligned}
&R_{\mathrm{surr}}(\hat{f}_{\mathrm{surr}}) - R_{\mathrm{surr}}(f^\star_{\mathrm{surr}}) \\
&\quad = \left(\widehat{R}_{\mathrm{surr}}(\hat{f}_{\mathrm{surr}}) - \widehat{R}_{\mathrm{surr}}(f^\star_{\mathrm{surr}})\right) + \left(R_{\mathrm{surr}}(\hat{f}_{\mathrm{surr}}) - \widehat{R}_{\mathrm{surr}}(\hat{f}_{\mathrm{surr}})\right) + \left(\widehat{R}_{\mathrm{surr}}(f^\star_{\mathrm{surr}}) - R_{\mathrm{surr}}(f^\star_{\mathrm{surr}})\right) \\
&\quad \leq \left(R_{\mathrm{surr}}(\hat{f}_{\mathrm{surr}}) - \widehat{R}_{\mathrm{surr}}(\hat{f}_{\mathrm{surr}})\right) + \left(\widehat{R}_{\mathrm{surr}}(f^\star_{\mathrm{surr}}) - R_{\mathrm{surr}}(f^\star_{\mathrm{surr}})\right) \\
&\quad \leq 2 \sup_{f \in \mathcal{F}} |\widehat{R}_{\mathrm{surr}}(f) - R_{\mathrm{surr}}(f)| \\
&\quad = 2 \sup_{\boldsymbol{g} \in \mathcal{G}} |\widehat{R}_{\mathrm{surr}}(\boldsymbol{g}) - R_{\mathrm{surr}}(\boldsymbol{g})| \\
&\quad \leq 4\mathfrak{R}_{n_{\mathrm{tr}}}(\ell \circ \mathcal{G}) + 2M_\ell \sqrt{\frac{\ln(2/\delta)}{2n_{\mathrm{tr}}}} \\
&\quad \leq \frac{8\sqrt{2}m\mathcal{L}_\ell}{\alpha^2} \mathfrak{R}_{n_{\mathrm{tr}}}(\mathcal{F}) + 2M_\ell \sqrt{\frac{\ln(2/\delta)}{2n_{\mathrm{tr}}}},
\end{aligned}$$

where the second equality is due to that $\mathcal{G} = \{\boldsymbol{x} \mapsto \boldsymbol{T}(f(\boldsymbol{x})) \mid f \in \mathcal{F}\}$ and $\boldsymbol{T}(\cdot)$ is deterministic. □

B. Supplementary Information on the Experiments

In this appendix, we provide supplementary information on the experiments.

B.1. Datasets

We describe details of the datasets as follows.

MNIST This is a dataset of normalized grayscale images containing handwritten digits from 0 to 9. All the images are fitted into a 28 × 28 pixels. The total number of training images and test images is 60,000 and 10,000 respectively. We use the even digits as the positive class and odd digits as the negative class.

Fashion-MNIST This is a dataset of grayscale images of different types of modern clothes. All the images are of the size 28 × 28 pixels. Similar to MNIST, this dataset has 60,000 training images and 10,000 test images. We convert this 10-class dataset into a binary dataset as follows:

- The classes 'Pullover', 'Dress', 'T-shirt', 'Trouser', 'Shirt', 'Bag', 'Ankle boot' and 'Sneaker' are denoted as the positive class;
- The classes 'Coat' and 'Sandal' are denoted as the negative class.

Kuzushiji-MNIST This is a dataset of grayscale images of cursive Japanese (Kuzushiji) characters. This dataset also has all images of size 28 × 28. And the total number of training images and test images is 60,000 and 10,000 respectively. We convert this 10-class dataset into a binary dataset as follows:

- The classes 'ki', 're', and 'wo' are denoted as the positive class;
- The classes 'o', 'su', 'tsu', 'na', 'ha', 'ma', and 'ya' are denoted as the negative class.

CIFAR-10 This dataset is made up of color images of ten types of objects and animals. The size of all images in this dataset is 32 × 32. There are 5,000 training images and 1,000 test images for each class, so 50,000 training and 10,000 test images in total. We convert this 10-class dataset into a binary dataset as follows:

- The positive class consists of 'airplane', 'bird', 'deer', 'dog', 'frog', 'cat', and 'horse';
- The negative class consists of 'automobile', 'ship', and 'truck'.

The generation of each U set is the same for all four benchmark datasets. More specifically, given the number of U sets m, class priors $\{\pi_j\}_{j=1}^{m}$, and the set sizes $\{n_j\}_{j=1}^{m}$, for j-th U set, we go through the following process:

1. Randomly shuffle the benchmark dataset;
2. Randomly select $n_j^p = n_j \times \pi_j$ samples of positive class;
3. Randomly select $n_j^n = n_j - n_j^p$ samples of negative class;
4. Combine them and we obtain the j-th U set.

B.2. Models

We describe details of the model architecture and optimization algorithm as follows.

MLP It is a 5-layer fully connected perceptron with ReLU (Nair & Hinton, 2010) as the activation function. The model architecture was $d - 300 - 300 - 300 - 1$, where d is the dimension of the input. Batch normalization (Ioffe & Szegedy, 2015) was applied before each hidden layer and ℓ_2-regularization was added. Dropout (Srivastava et al., 2014) with rate 0.2 was also added before each hidden layer. The optimizer was Adam (Kingma & Ba, 2014) with the default momentum parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$).

ResNet-32 It is a 32-layer residual network (He et al., 2016) and the architecture was as follows:
0th (input) layer: $(32 * 32 * 3)-$
1st to 11th layers: $C(3 * 3, 16) - [C(3 * 3, 16), C(3 * 3, 16)] * 5-$
12th to 21st layers: $[C(3 * 3, 32), C(3 * 3, 32)] * 5-$
22nd to 31st layers: $[C(3 * 3, 64), C(3 * 3, 64)] * 5-$

32nd layer: Global Average Pooling-1,
where $C(3*3,96)$ represents a 96-channel of $3*3$ convolutions followed by a ReLU activation function, $[\cdot]$*2 represents a repeat of twice of such layer, $C(3*3,96,2)$ represents a similar layer but with stride 2, and $[\cdot,\cdot]$ represents a building block. Batch normalization was applied for each hidden layers and ℓ_2-regularization was also added. The optimizer was Adam with the default momentum parameters ($\beta_1 = 0.9$ and $\beta_2 = 0.999$).

The MLP model was used for the MNIST, Fashion-MNIST, Kuzushiji-MNIST dataset, and the ResNet-32 model was used for the CIFAR-10 dataset.

B.3. Other Details

We implemented all the methods by Keras and conducted all the experiments on an NVIDIA Tesla P100 GPU. The batch size was 256 for all the methods. For MNIST, Fashion-MNIST, and Kuzushiji MNIST dataset, the initial learning-rate was 1e-5 for U^m-SSC and 1e-4 for the MMC based methods and LLP-VAT. For CIFAR-10 dataset, the initial learning-rate was 5e-6 for U^m-SSC and 1e-5 for the MMC based methods and LLP-VAT. In addition, the learning rate was decreased by $1/(1+\text{decay}\cdot\text{epoch})$, where the decay parameter was 1e-4. This is the built-in learning rate scheduler of Keras.

We describe details of the hyper-parameters for the baseline methods as follows.

- MMC-U^2-b (Scott & Zhang, 2020): by assuming that the number of sets $m = 2k$, this baseline method firstly pairs all the U sets and then linearly combines the unbiased balanced risk estimator of each pair, The learning objective is

$$\widehat{R}_{\text{MMC-U}^2\text{-b}}(f) = \sum\nolimits_{j=1}^{k} \omega_j \widehat{R}_{\text{U}^2\text{-b}}(f),$$

where

$$\begin{aligned}\widehat{R}_{\text{U}^2\text{-b}}(f) = \frac{c_{b1}^{+}}{n}\sum_{i=1}^{n_1}\ell_b(f(\boldsymbol{x}_i^1),+1) - \frac{c_{b2}^{+}}{n}\sum_{j=1}^{n_2}\ell_b(f(\boldsymbol{x}_j^2),+1)\\ - \frac{c_{b1}^{-}}{n}\sum_{i=1}^{n_1}\ell_b(f(\boldsymbol{x}_i^1),-1) + \frac{c_{b2}^{-}}{n}\sum_{j=1}^{n_2}\ell_b(f(\boldsymbol{x}_j^2),-1),\end{aligned}$$

$c_{b1}^{+} = \frac{1-\pi_2}{2(\pi_1-\pi_2)}$, $c_{b1}^{-} = \frac{\pi_2}{2(\pi_1-\pi_2)}$, $c_{b2}^{+} = \frac{1-\pi_1}{2(\pi_1-\pi_2)}$, and $c_{b2}^{-} = \frac{\pi_1}{2(\pi_1-\pi_2)}$. For the pairing process, since we use the uniform set sizes, i.e., the set size of each U set is the same as n_{tr}/m ($n_{\text{tr}} = 60,000$ in MNIST, Fashion-MNIST, and Kuzushiji-MNIST, $n_{\text{tr}} = 50,000$ in CIFAR-10), we pair all the U sets following Proposition 9 in Appendix S6 of Scott & Zhang (2020), i.e., match the U set with the largest class prior π_j with the smallest, the U set with the second largest class prior π_j with the second smallest, and so on. For the combination weights, we set them following Theorem 5 in Section 2.2 of Scott & Zhang (2020). More specifically, for the j-th pair of U sets: $\mathcal{X}_{\text{tr}}^1$ and $\mathcal{X}_{\text{tr}}^2$, assume $\pi_1 > \pi_2$, since we use uniform set sizes, the optimal weights $\omega_j \propto (\pi_1 - \pi_2)^2$. So we set the weight ω_j as $(\pi_1-\pi_2)^2$ and then normalize all of them to sum to 1, i.e., $\sum_{j=1}^{k}\omega_j = 1$.
- MMC-U^2: this method improves the MMC-U^2-b baseline by replacing the unbiased balanced risk estimator $\widehat{R}_{\text{U}^2\text{-b}}(f)$ with the unbiased risk estimators $\widehat{R}_{\text{U}^2}(f)$ (Lu et al., 2019). The learning objective is

$$\widehat{R}_{\text{MMC-U}^2}(f) = \sum\nolimits_{j=1}^{k} \omega_j \widehat{R}_{\text{U}^2}(f),$$

where

$$\begin{aligned}\widehat{R}_{\text{U}^2}(f) = \underbrace{\frac{c_{1}^{+}}{n}\sum_{i=1}^{n_1}\ell_b(f(\boldsymbol{x}_i^1),+1) - \frac{c_{2}^{+}}{n}\sum_{j=1}^{n_2}\ell_b(f(\boldsymbol{x}_j^2),+1)}_{\widehat{R}_{\text{U}^2\text{-p}}(f)}\\ \underbrace{-\frac{c_{1}^{-}}{n}\sum_{i=1}^{n_1}\ell_b(f(\boldsymbol{x}_i^1),-1) + \frac{c_{2}^{-}}{n}\sum_{j=1}^{n_2}\ell_b(f(\boldsymbol{x}_j^2),-1),}_{\widehat{R}_{\text{U}^2\text{-n}}(f)}\end{aligned}$$

$c_1^+ = \frac{(1-\pi_2)\pi_\mathcal{D}}{\pi_1-\pi_2}$, $c_1^- = \frac{\pi_2(1-\pi_\mathcal{D})}{\pi_1-\pi_2}$, $c_2^+ = \frac{(1-\pi_1)\pi_\mathcal{D}}{\pi_1-\pi_2}$, and $c_2^- = \frac{\pi_1(1-\pi_\mathcal{D})}{\pi_1-\pi_2}$. The pairing process and the combination weights setup follow those of MMC-U^2-b.

- MMC-U^2-c: this method improves the MMC-U^2 baseline by replacing the unbiased risk estimators $\widehat{R}_{\mathrm{U}^2}(f)$ with the non-negative risk estimators $\widehat{R}_{\mathrm{U}^2\text{-c}}(f)$ (Lu et al., 2020). The learning objective is

$$\widehat{R}_{\text{MMC-U}^2\text{-c}}(f) = \sum\nolimits_{j=1}^{k} \omega_j \widehat{R}_{\mathrm{U}^2\text{-c}}(f),$$

where

$$\widehat{R}_{\mathrm{U}^2\text{-c}}(f) = f_\mathrm{c}(\widehat{R}_{\mathrm{U}^2\text{-p}}(f)) + f_\mathrm{c}(\widehat{R}_{\mathrm{U}^2\text{-n}}(f)).$$

According to Lu et al. (2020), the *generalized leaky ReLU* function, i.e.,

$$f_\mathrm{c}(r) = \begin{cases} r & (r \geq 0), \\ -\kappa r & (r < 0), \end{cases}$$

for $\kappa \geq 0$, works well as the correction function f_c, so we choose it for implementing this baseline method. The hyper-parameter κ was chosen based on a validation dataset, and the pairing process and the combination weights setup follow those of MMC-U^2-b.

- LLP-VAT (Tsai & Lin, 2020): this baseline method is based on empirical proportion risk minimization. The learning objective is

$$\widehat{R}_{\text{prop-c}}(f) = \widehat{R}_{\text{prop}}(f) + \alpha \ell_{\text{cons}}(f),$$

where

$$\widehat{R}_{\text{prop}}(f) = \sum\nolimits_{j=1}^{m} d_{\text{prop}}(\pi_j, \hat{\pi}_j)$$

is the proportion risk, π_j and

$$\hat{\pi}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} \frac{1 + \operatorname{sign}(f(\boldsymbol{x}_i^j) - 1/2)}{2}$$

are the true and predicted label proportions for the j-th U set $\mathcal{X}_{\text{tr}}^j$, d_{prop} is a distance function, and

$$\ell_{\text{cons}}(f) = d_{\text{cons}}(f(\boldsymbol{x}), f(\hat{\boldsymbol{x}}))$$

is the consistency loss, d_{cons} is a distance function, $\hat{\boldsymbol{x}}$ is a perturbed input from the original one $\boldsymbol{x}$. We set the hyper-parameters $\alpha = 0.05$ and the perturbation weight $\mu = 6.0$ for LLP-VAT following the default implementation in their paper (Tsai & Lin, 2020).

C. Supplementary Experimental Results

In this appendix, we provide supplementary experimental results.

C.1. Comparison with State-of-the-art Methods

Please find Table 5 the final classification errors of comparing our proposed method with state-of-the-art methods on learning from 10, 25, and 50 U sets (corresponds to Figure 2).

In the experiments, we also find that the empirical training risk of the proposed U^m-SSC is obviously higher than all other baseline methods. This is due to the added transition layer and the rescales the output range. We provide a detailed explanation as follows.

By using the monotonicity of the transition function $T_j(\cdot)$ (Menon et al., 2015), we can compute the range of the model output. Since $g(\boldsymbol{x}) \in [0, 1]$, by plugging in $g(\boldsymbol{x}) = 0$ and $g(\boldsymbol{x}) = 1$ respectively we obtain

$$T_j(0) = \frac{b_j}{d} = \frac{\rho_j \pi_\mathcal{D}(1-\pi_j)}{\sum_{j=1}^{m} \rho_j \pi_\mathcal{D}(1-\pi_j)},$$

$$T_j(1) = \frac{a_j + b_j}{c_j + d} = \frac{\rho_j \pi_j(1-\pi_\mathcal{D})}{\sum_{j=1}^{m} \rho_j \pi_j(1-\pi_\mathcal{D})}.$$

Table 5. Means (standard deviations) of the classification error over three trials in percentage of each method on learning from 10, 25 and 50 U sets. Best and comparable methods (paired *t*-test at significance level 5%) are highlighted in boldface.

Dataset	Sets	MMC-U^2-b	MMC-U^2	MMC-U^2-c	LLP-VAT	U^m-SSC
MNIST	10	7.7(0.55)	8.03(0.74)	4.46(0.23)	3.62(0.38)	**3.05(0.08)**
	25	5.35(0.22)	5.32(0.28)	3.69(0.11)	3.28(0.35)	**2.51(0.02)**
	50	5.81(0.22)	5.82(0.12)	3.29(0.09)	3.02(0.22)	**2.86(0.04)**
Fashion-MNIST	10	16.63(1.38)	9.49(0.37)	8.12(0.51)	21.23(3.52)	**6.5(0.21)**
	25	11.1(0.45)	9.12(0.1)	7.45(0.1)	26.66(0.4)	**6.14(0.02)**
	50	11.18(0.53)	9.6(0.47)	8.52(0.48)	27.92(2.22)	**6.6(0.06)**
Kuzushiji-MNIST	10	16.25(0.61)	15.23(0.3)	12.88(0.35)	16.12(0.41)	**9.83(0.4)**
	25	15.93(0.71)	14.02(0.12)	10.18(0.33)	19.48(1.84)	**8.98(0.07)**
	50	15.8(0.37)	12.46(0.43)	9.69(0.37)	18.94(0.4)	**8.97(0.52)**
CIFAR-10	10	15.83(0.21)	16.01(0.32)	14.33(0.06)	19.38(0.05)	**13.43(0.14)**
	25	19.6(0.77)	16.18(0.27)	14.19(0.25)	16.89(0.15)	**13.31(0.13)**
	50	21.1(1.03)	16.08(0.38)	14.28(0.13)	17.66(0.57)	**13.32(0.19)**

Table 6. Means (standard deviations) of the classification error over three trials in percentage for the U^m-SSC method tested on inaccurate class priors.

Dataset	Sets	True	$\epsilon = 0.05$	$\epsilon = 0.1$	$\epsilon = 0.15$	$\epsilon = 0.2$
MNIST	10	2.54(0.02)	2.64(0.06)	3.31(0.18)	2.98(0.14)	3.84(0.25)
	50	2.45(0.04)	2.52(0.02)	2.69(0.04)	3.11(0.19)	3.16(0.13)
Fashion-MNIST	10	6.22(0.05)	6.31(0.03)	6.13(0.13)	6.61(0.04)	9.39(0.19)
	50	6.37(0.26)	6.39(0.17)	6.76(0.11)	7.64(0.22)	10.91(0.47)
Kuzushiji-MNIST	10	8.74(0.24)	8.97(0.23)	9.77(0.29)	11.31(0.21)	11.62(0.56)
	50	9.0(0.22)	9.27(0.26)	9.15(0.15)	9.38(0.18)	10.61(0.03)
CIFAR-10	10	13.54(0.23)	13.7(0.25)	14.43(0.22)	16.82(0.29)	19.7(0.54)
	50	13.55(0.18)	13.75(0.09)	14.19(0.22)	15.69(0.21)	18.84(0.26)

According to our generation processes of class priors and set size, $\pi_j \in [0.1, 0.9]$ and $\rho_j = 1/m$ for any $j = 1, \ldots, m$. The upper bound of the model output $\max(T_j(0), T_j(1))$ takes value between 0.01 and 0.1. As a result, the cross-entropy loss gives its value in range $[2.3, 4.6]$, which is relatively high than usual training loss. We note that this high training loss has an effect on hyper-parameters tuning, especially for the learning rate. We may need a relatively small learning rate for better performance of our method.

C.2. Robustness against Inaccurate Class Priors

Please find Table 6 the final classification errors of our method on learning from 50 U sets with inaccurate class priors (corresponds to Figure 3).