Fairness-Aware Online Positive-Unlabeled Learning

Hoin Jung, Xiaoqian Wang
Elmore Family School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907
{jung414, joywang}@purdue.edu

Abstract

Machine learning applications for text classification are increasingly used in domains such as toxicity and misinformation detection in online settings. However, obtaining precisely labeled data for training remains challenging, particularly because not all problematic instances are reported. Positive-Unlabeled (PU) learning, which uses only labeled positive and unlabeled samples, offers a solution for these scenarios. A significant concern in PU learning, especially in online settings, is fairness: specific groups may be disproportionately classified as problematic. Despite its importance, this issue has not been explicitly addressed in research. This paper aims to bridge this gap by investigating the fairness of PU learning in both offline and online settings. We propose a novel approach to achieve more equitable results by extending PU learning methods to online learning for both linear and non-linear classifiers and analyzing the impact of the online setting on fairness. Our approach incorporates a convex fairness constraint during training, applicable to both offline and online PU learning. Our solution is theoretically robust, and experimental results demonstrate its efficacy in improving fairness in PU learning in text classification.

1 Introduction

A classification system with machine learning for text data has been developed widely for various application such as toxicity classification (Thain et al., 2017; Wulczyn et al., 2017; Androcec, 2020; Li et al., 2022b) and misinformation detection (Go et al., 2022; Park et al., 2022). However, obtaining precisely labeled data for training can be an arduous task (Du Plessis et al., 2015), and the absence of positivity does not automatically equate to negativity in some cases (Hsieh et al., 2015). For example, in both toxicity and misinformation detection, only part of textual contents containing toxicity or misinformation are reported as concerns, as illustrated in Fig.1. *Positive-unlabeled (PU) learning* (Elkan and Noto, 2008; Du Plessis et al., 2015; Kiryo et al., 2017) aims to learn from this incomplete information and achieve reliable classification by using only labeled-positive and unlabeled samples, where the unlabeled samples are permitted to be classified as either positive or negative.

Furthermore, we acknowledge the necessity of an online learning framework in PU learning. Firstly, integrating PU learning with online learning can effectively address real-world challenges (Zhang et al., 2021), when a machine learning system operates in dynamic environments where new data is continuously arriving. For example, as visualized in Fig.1, the patterns of toxicity or misinformation evolve online, so the machine learning system needs to keep training on newly arrived data with new patterns, while only a few documents are reported as concerns. However, offline batch training is inadequate to sequentially provided data, as retraining the system from scratch with all the data is costly (Thennakoon et al., 2019), while unreported cases might also possess the potential for positivity (de Souza et al., 2022), necessitating the utilization of a PU learning framework in online scenario (Zhang et al., 2021).

However, PU learning faces a significant fairness issue by disproportionately predicting certain groups as positive based on factors such as gender, race, and the presence of specific features. Fairness concerns in PU learning stem from two different perspectives. First, the training data might naturally contain biases. For example, in the Wikipedia Talk dataset (Thain et al., 2017; Wulczyn et al., 2017) for toxicity classification, 36.03% of documents with sexuality terms contain toxicity, whereas only 9.28% of documents without sexuality terms are toxicity documents. A PU learning-based automated toxicity classification system might overly depend on the existence of sexuality terms, resulting in unfair predictions by misleading to an in-

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 170–185
November 12-16, 2024 ©2024 Association for Computational Linguistics

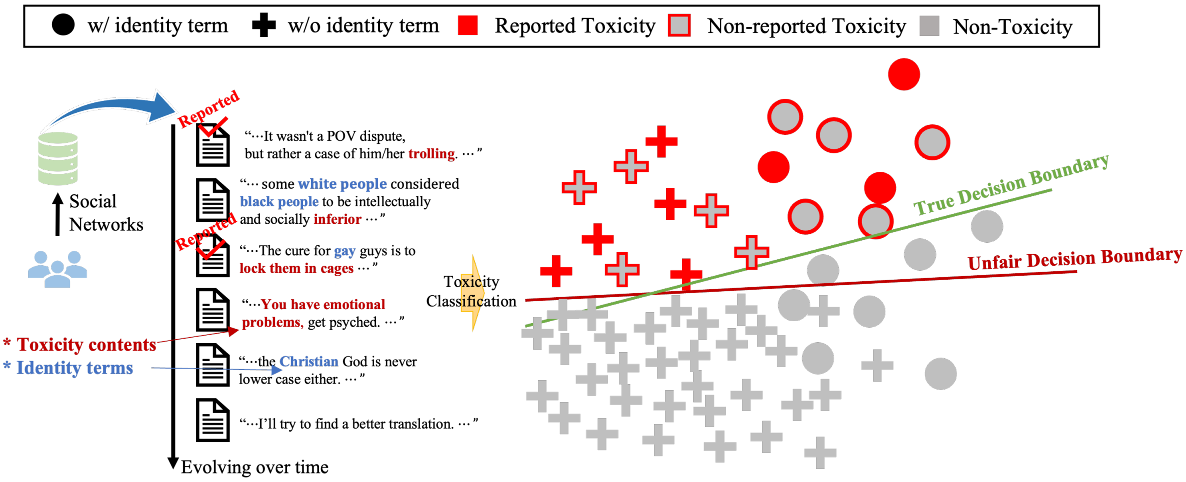

Figure 1: Online Positive-Unlabeled (PU) learning is an effective framework for toxicity classification in social networks, where only a subset of positive (reported toxicity) samples are labeled and there exist unlabeled positive (non-reported toxicity), and the data pool evolves over time. However, online PU learning may encounter fairness challenges due to prevalent biases in the data, where contents with identity terms have higher chances of toxicity compared to those without identity terms, as well as the long-term constraints inherent in online learning.

creased false positive rate (FPR). Secondly, PU learning tends to produce a higher false positive rate because the PU framework is inherently blind in differentiating false positives from false negatives due to the lack of negative samples in the unlabeled pool (Kong et al., 2019). To address this issue, the risk estimator for PU learning tends to convert negative risk to unlabeled risk based on the class prior (Du Plessis et al., 2015), which compulsively assigns positive labels to a portion of unlabeled data, resulting in higher FPR, as illustrated in Fig.2 (b). Despite its relevance, the fairness issue in PU learning remains largely unexplored (Wu and He, 2022), and existing fairness literature (Jang et al., 2021; Chai and Wang, 2022) is mostly confined to PN learning, where all labels are readily available.

Furthermore, online learning may encounter fairness issues due to its long-term constraint (Zhao et al., 2021). The original data's uneven distribution across sensitive groups means each incremental stage might have few or no samples from certain subgroups, especially with limited positive samples. Such imbalances can reduce diversity in each incremental stage. The incrementally provided data may not accurately reflect the overall distribution, potentially leading to a higher false positive rate and more unfair predictions. In Fig.2 (c), the comparison between the solid bar (offline) and the hatched bar (online) demonstrates that online learning can worsen fairness issues in PU learning.

In short, online PU learning suffers from twofold fairness violations due to both **1) PU learning** and **2) online learning**. Wu and He (2022) first addressed fairness in PU learning, but the reason of bias in PU learning was not extensively studied. Additionally, Wu and He (2022) relies on the selected completely at random (SCAR) assumption (Elkan and Noto, 2008), which could be unrealistic in practice. Zhang et al. (2021) proposed an online PU learning framework, but it didn't discuss fairness issue in PU learning and was limited to linear classifiers. Overall, fairness in offline PU learning is largely unexplored, and no research explicitly addresses fairness in online PU learning, making it a pressing concern.

In this paper, we firstly address this gap by studying fairness in PU learning and extend it to the online framework by introducing a convex fairness constraint ensuring Equalized Odds fairness, while maintaining the model's prediction capacity. Specifically, we apply PU learning methods to online learning for linear, Multilayer Perceptron (MLP), and Long Short-Term Memory (LSTM) (Graves and Graves, 2012) classifiers, analyzing the impact of the online setting on fairness by defining the concept of *fair regret*. Our proposed approach, *fairness-aware online positive-unlabeled learning* (FOPU) is theoretically grounded, and we provide experimental results to demonstrate its efficacy in enhancing fairness in PU learning. To this end, this paper offers a practical framework for implement-

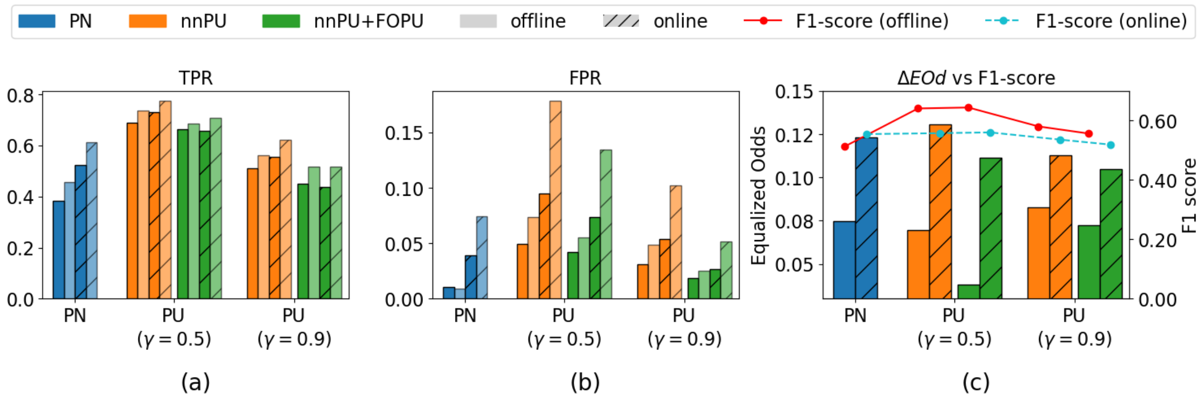

Figure 2: In the Wiki Toxicity dataset, we compare scenarios with (green) and without (orange) a fairness constraint, using LSTM classifier. Bar plots illustrate the True Positive Rate (TPR), False Positive Rate (FPR), and ΔEOd, while line plots show F1-score. In the first two subfigures, darker bars represent a document group without sexuality term, and lighter bars correspond to a group with sexuality term. Bars with hatching indicate online learning. The figure reveals that both PU learning (orange) and online learning (hatched) result in a higher FPR compared to PN learning (blue) and offline learning (solid), respectively. Implementing fairness-aware training (green) reduces the disparity in the FPR between demographic groups, thereby promoting fairness while preserving F1-score.

ing fairer online learning applications for text classification across various real-world contexts. We validate the effectiveness of our approach through extensive experimental results, ensuring fairness without compromising its utility, i.e., F1-score.

2 Related Work

Fairness. To achieve fairness in classification tasks, diverse methodologies have been proposed. These include pre-processing, post-processing, and in-processing approaches. Pre-processing approaches focus on refining training data such as data reweighing (Chai and Wang, 2022; Li and Liu, 2022) and data augmentation (Jang et al., 2021; Rajabi and Garibay, 2022). Based on the ordinarily trained classifier, post-processing methods optimize the accuracy-fairness trade-off using confusion matrix (Kim et al., 2020) or manipulating threshold (Jang et al., 2022). In-processing methods directly incorporate fairness constraints into the learning algorithm itself making the model explicitly learn a desired fairness criteria (Zafar et al., 2017b,a). Particularly, making the fairness constraint convex is important since it ensures the existence of a unique optimal solution. Wu et al. (2019) suggested a relaxed convex fairness constraint as an objective function to be optimized.

Positive-Unlabeled learning. Elkan and Noto (2008) assumes that labeled examples are selected completely at random (SCAR) from the entire body of positive samples. However, the assumption of SCAR is unrealistic in practice (Bekker and Davis, 2020), and overestimates the true class prior (Christoffel et al., 2016). Du Plessis et al. (2015) and Kiryo et al. (2017) suggested optimizing PU risk estimators using true class prior by converting the negative empirical risk to unlabeled empirical risk. Moreover, various types of PU frameworks are suggested utilizing label distribution (Kato et al., 2019; Zhao et al., 2022), data-reweighing (Zhu et al., 2023), and data augmentation (Li et al., 2022a).

Online Learning. Online Gradient Descent (OGD) (Zinkevich, 2003) is a fundamental technique in online learning, while only linear classifier is considered in (Zinkevich, 2003). Sahoo et al. (2017) suggested Online Deep Learning making online learning for a neural network. In this paper, we apply the same strategy (Sahoo et al., 2017) to make LSTM (Graves and Graves, 2012) online.

Composite Task. Fairness in machine learning, positive-unlabeled learning, and online learning are three distinct yet deeply interconnected fields. Zhao et al. (2021) and Patil et al. (2021) discussed fairness in online learning but not in real-time manner. Zhang et al. (2021) proposed online PU learning, viable only for linear classifiers, but the fairness concern is not discussed. Although Wu and He (2022) suggested a post-processing framework attaining fairness in PU learning, it is based on SCAR assumption which is impractical (Bekker and Davis, 2020), and not feasible to online learning framework and PU risk estimators.

3 Method

3.1 Risk Estimator for PU learning

In PU learning, instead of the class label $y \in \{+1, -1\}$, we use the label indicator $s \in \{+1, -1\}$, $s = +1$ denoting the label exists and the class of the sample is positive, while $s = -1$ indicates the label does not exist and the class of the sample can be either positive or negative.

Denote the class-conditional densities for positive and negative class as $p_p(\boldsymbol{x}) = p(\boldsymbol{x}|y = +1)$, and $p_n(\boldsymbol{x}) = p(\boldsymbol{x}|y = -1)$ where $\boldsymbol{x} \in \mathbb{R}^d$ is input data, and $y \in \{+1, -1\}$ is the binary class label. Also, let $p(\boldsymbol{x})$ denote the marginal density regarding unlabeled data. Then, $p(x) = \pi p_p(\boldsymbol{x}) + (1 - \pi)p_n(\boldsymbol{x})$ if we assume that the positive class-prior probability $\pi = p(y = +1)$ and $p(y = -1) = 1 - \pi$ are given. In the positive-negative (PN) setting, we minimize the following risk estimator for a real-valued classifier $\hat{y} = sign(f(\boldsymbol{x}))$, $f : \mathcal{X} \to \mathbb{R}$,

$$R_{\text{pn}}(f) = \pi \mathbb{E}_p[\ell(f(\boldsymbol{X}))] + (1 - \pi)\mathbb{E}_n[\ell(-f(\boldsymbol{X}))]$$

where $\mathbb{E}_p[\cdot] = \mathbb{E}_{\boldsymbol{X} \sim p_p(x)}$ and $\mathbb{E}_n[\cdot] = \mathbb{E}_{\boldsymbol{X} \sim p_n(x)}$, and ℓ is a surrogate loss function such as square loss, zero-one loss, and double hinge loss. Based on the fact that $p(\boldsymbol{x}) = \pi p(\boldsymbol{x}|y = +1) + (1 - \pi)p(\boldsymbol{x}|y = -1)$, the 'negative' risk can be replaced with 'unlabeled' risk such that

$$\mathbb{E}_u[\ell(-f(\boldsymbol{X}))] = \pi \mathbb{E}_p[\ell(-f(\boldsymbol{X}))] + (1 - \pi)\mathbb{E}_n[\ell(-f(\boldsymbol{X}))]$$

Therefore, the risk estimator for PU learning (Du Plessis et al., 2015) can be approximated by

$$R_{\text{upu}}(f) = \pi \mathbb{E}_p[\ell(f(\boldsymbol{X}))] + \Big[\mathbb{E}_u[\ell(-f(\boldsymbol{X}))] - \pi \mathbb{E}_p[\ell(-f(\boldsymbol{X}))]\Big]. \tag{1}$$

Furthermore, we adopt nnPU (Kiryo et al., 2017). nnPU is modified version of uPU to prevent overfitting to training data,

$$R_{\text{nnpu}}(f) = \pi \mathbb{E}_p[\ell(f(\boldsymbol{X}))] + \max\Big(0, \Big[\mathbb{E}_u[\ell(-f(\boldsymbol{X}))] - \pi \mathbb{E}_p[\ell(-f(\boldsymbol{X}))]\Big]\Big).$$

However, PU learning suffers from fairness issues as described in Fig.2 and Appendix A by posing higher FPR. To this end, we propose the need for a fairness constraint on PU learning and its impact on prediction in the following sections.

3.2 Fairness Constraints and Convexity

In this paper, we utilize a fairness constraints such as the Difference of Demographic Parity (DP) and Difference of Equalized Odds (EOd). DP requires independence between the predicted outcome and the sensitive information $a \in \{+1, -1\}$, $P(\hat{y}|a = -1) = P(\hat{y}|a = +1)$, i.e. $\hat{y} \perp\!\!\!\perp a$. However, the usefulness of DP is limited to cases where there exists a correlation between y and a such that $y \not\perp\!\!\!\perp a$. EOd overcomes the limitation of DP by conditioning the metric on the ground truth Y, i.e. $P(\hat{y}|a = +1, y) = P(\hat{y}|a = -1, y), \forall y \in \{+1, -1\}$. Based on convex form of DP suggested in (Wu et al., 2019), we extend the convex fairness constraint for EOd. DP and EOd will be used as evaluation metrics to verify each model's performance, while EOd convex form is used as a part of the objective function. Details of fairness constraints are introduced in Eq.(3) and Appendix B.

3.3 Fairness-aware Online PU learning

We propose a fairness-aware PU learning framework for both offline and online learning. Specifically, we use Lagrangian relaxation such that

$$\mathcal{R}_{\text{off}}(f) = R_{\text{pu}}(f) + \lambda_r \Omega(f) + \lambda_f R_{\text{fair}}(f) \tag{2}$$

where λ_r and λ_f are hyperparameters, $R_{\text{pu}}(f)$ can be any PU risk estimator, and $R_{\text{fair}}(f)$ is the fairness constraints. In detail, in the training step, $R_{\text{fair}}(f)$ is determined by the sign of the empirical fairness measure in every iteration,

$$R_{\text{fair}}(f) = \begin{cases} EOd_\kappa(f) & \text{if } EOd(f) \geq 0 \\ EOd_\delta(f) & \text{if } EOd(f) < 0, \end{cases} \tag{3}$$

where

$$EOd_\kappa(f) = \mathbb{E}\Big[\tfrac{\mathbb{I}_{a=1,y=1}}{p_{1,1}}\kappa(f(x)) - \big(1 - \tfrac{\mathbb{I}_{a=-1,y=1}}{\pi - p_{1,1}}\kappa(-f(x))\big)\Big]$$
$$+\mathbb{E}\Big[\tfrac{\mathbb{I}_{a=1,y=-1}}{p_{1,-1}}\kappa(f(x)) - \Big(1 - \tfrac{\mathbb{I}_{a=-1,y=-1}}{1-\pi-p_{1,-1}}\kappa(-f(x))\Big)\Big]$$

$$EOd_\delta(f) = \mathbb{E}\Big[\tfrac{\mathbb{I}_{a=1,y=1}}{p_{1,1}}\delta(f(x)) - \big(1 - \tfrac{\mathbb{I}_{a=-1,y=1}}{\pi - p_{1,1}}\delta(-f(x))\big)\Big]$$
$$+\mathbb{E}\Big[\tfrac{\mathbb{I}_{a=1,y=-1}}{p_{1,-1}}\delta(f(x)) - \big(1 - \tfrac{\mathbb{I}_{a=-1,y=-1}}{1-\pi-p_{1,-1}}\delta(-f(x))\big)\Big]$$

are convex form of EOd fairness constraints where κ is a convex surrogate function $\kappa(z) = \max(z + 1, 0)$ and δ is a concave surrogate function $\delta(z) = \min(z, 1)$. However, $R_{\text{fair}}(f)$ potentially reduces the TPR to achieve equalized TPR across the group. To prevent a reduction in TPR, we apply a *penalty term* to $R_{\text{fair}}(f)$ when the empirical TPR is lower or FPR is higher than in the previous iteration. Details and its impact are in Appendix B.3 and B.4.

For online learning, we consider multiple data $I_t = \{(\boldsymbol{x}_t^{(i)}, y_t^{(i)})\}_{i=1}^b$ is provided at round t ($t =$

Wiki		Baseline F1-score	Baseline ΔDP	Baseline ΔEOd	Fairness-aware Learning F1-score	Fairness-aware Learning ΔDP	Fairness-aware Learning ΔEOd
Linear	uPU	0.5485 ± 0.0033	0.2618 ± 0.0079	0.1721 ± 0.0156	**0.5622 ± 0.0038**	**0.2216 ± 0.0089**	**0.1620 ± 0.0175**
	nnPU	0.5491 ± 0.0034	0.2628 ± 0.0080	0.1738 ± 0.0176	**0.5609 ± 0.0035**	**0.2191 ± 0.0073**	**0.1575 ± 0.0128**
MLP	uPU	0.5940 ± 0.0109	0.2262 ± 0.0118	0.0934 ± 0.0253	**0.6033 ± 0.0094**	**0.2188 ± 0.0192**	**0.0798 ± 0.0163**
	nnPU	0.5544 ± 0.0285	0.2237 ± 0.0174	0.0859 ± 0.0238	**0.5849 ± 0.0105**	**0.2158 ± 0.0098**	**0.0589 ± 0.0155**
LSTM	uPU	0.6019 ± 0.0190	**0.1684 ± 0.0142**	0.0860 ± 0.0222	**0.6216 ± 0.0097**	0.1710 ± 0.0152	**0.0558 ± 0.0191**
	nnPU	0.6400 ± 0.0063	0.2031 ± 0.0114	0.0697 ± 0.0170	**0.6433 ± 0.0056**	**0.1823 ± 0.0145**	**0.0382 ± 0.0204**
Chat Toxicity		Baseline F1-score	Baseline ΔDP	Baseline ΔEOd	Fairness-aware Learning F1-score	Fairness-aware Learning ΔDP	Fairness-aware Learning ΔEOd
Linear	uPU	**0.4013 ± 0.0134**	0.4106 ± 0.1104	0.4569 ± 0.1986	0.3912 ± 0.0142	**0.3158 ± 0.0665**	**0.3128 ± 0.1135**
	nnPU	**0.4013 ± 0.0075**	0.4599 ± 0.0798	0.5208 ± 0.1677	0.3874 ± 0.0112	**0.3254 ± 0.0498**	**0.3002 ± 0.0785**
MLP	uPU	**0.4145 ± 0.0251**	0.2758 ± 0.0967	0.2494 ± 0.1202	0.3666 ± 0.0209	**0.2334 ± 0.0602**	**0.1954 ± 0.0926**
	nnPU	**0.4272 ± 0.0279**	0.4003 ± 0.0847	0.4026 ± 0.1340	0.4178 ± 0.0280	**0.2740 ± 0.0859**	**0.2830 ± 0.1045**
LSTM	uPU	**0.4714 ± 0.0145**	0.2804 ± 0.0831	0.2734 ± 0.0878	0.4592 ± 0.0139	**0.2235 ± 0.0729**	**0.1827 ± 0.1258**
	nnPU	**0.4891 ± 0.0099**	0.3533 ± 0.0936	0.3136 ± 0.1748	0.4710 ± 0.0140	**0.2455 ± 0.0502**	**0.2075 ± 0.0983**
NELA		Baseline F1-score	Baseline ΔDP	Baseline ΔEOd	Fairness-aware Learning F1-score	Fairness-aware Learning ΔDP	Fairness-aware Learning ΔEOd
Linear	uPU	0.7780 ± 0.0022	0.0822 ± 0.0057	0.0549 ± 0.0097	**0.7849 ± 0.0009**	**0.0787 ± 0.0086**	**0.0469 ± 0.0182**
	nnPU	0.7781 ± 0.0021	0.0821 ± 0.0056	0.0551 ± 0.0095	**0.7855 ± 0.0013**	**0.0760 ± 0.0127**	**0.0497 ± 0.0158**
MLP	uPU	0.7710 ± 0.0042	0.1219 ± 0.0120	0.0422 ± 0.0225	**0.8029 ± 0.0079**	**0.1014 ± 0.0453**	**0.0406 ± 0.0247**
	nnPU	0.7919 ± 0.0029	**0.0653 ± 0.0312**	0.0379 ± 0.0253	**0.7961 ± 0.0044**	0.0866 ± 0.0091	**0.0222 ± 0.0153**
LSTM	uPU	0.7902 ± 0.0041	0.1283 ± 0.0111	0.1633 ± 0.0273	**0.8057 ± 0.0056**	**0.1006 ± 0.0110**	**0.0731 ± 0.0306**
	nnPU	**0.8041 ± 0.0055**	0.0867 ± 0.0240	0.1117 ± 0.0266	0.8010 ± 0.0028	**0.0497 ± 0.0188**	**0.0359 ± 0.0084**

Table 1: Experimental results for **offline** learning with and without fairness constraints. The superior results (higher F1-score; lower ΔDP and ΔEOd) for each evaluation metric are **bolded** for each combination of model, PU method, and dataset, comparing the baseline without fairness constraints to the model with fairness constraints.

Wiki		Baseline F1-score	Baseline ΔDP	Baseline ΔEOd	Fairness-aware Learning F1-score	Fairness-aware Learning ΔDP	Fairness-aware Learning ΔEOd
Linear	uPU	**0.5667 ± 0.0019**	0.2405 ± 0.0064	0.1740 ± 0.0113	0.5601 ± 0.0026	**0.2132 ± 0.0103**	**0.1506 ± 0.0209**
	nnPU	0.5625 ± 0.0030	0.2435 ± 0.0068	0.1734 ± 0.0112	**0.5633 ± 0.0020**	**0.2220 ± 0.0142**	**0.1531 ± 0.0221**
MLP	uPU	0.5424 ± 0.0057	0.2613 ± 0.0093	0.1707 ± 0.0214	**0.5544 ± 0.0076**	**0.2322 ± 0.0134**	**0.1505 ± 0.0202**
	nnPU	0.5421 ± 0.0086	0.2604 ± 0.0078	0.1714 ± 0.0220	**0.5545 ± 0.0073**	**0.2290 ± 0.0115**	**0.1463 ± 0.0201**
LSTM	uPU	**0.5617 ± 0.0130**	0.2170 ± 0.0239	0.1331 ± 0.0217	0.5583 ± 0.0080	**0.2034 ± 0.0200**	**0.1107 ± 0.0247**
	nnPU	**0.5570 ± 0.0086**	0.2400 ± 0.0180	0.1306 ± 0.0220	0.5507 ± 0.0178	**0.2246 ± 0.0224**	**0.1168 ± 0.0252**
Chat Toxicity		Baseline F1-score	Baseline ΔDP	Baseline ΔEOd	Fairness-aware Learning F1-score	Fairness-aware Learning ΔDP	Fairness-aware Learning ΔEOd
Linear	uPU	0.4070 ± 0.0353	0.3773 ± 0.1369	0.4977 ± 0.3522	**0.4423 ± 0.0229**	**0.3613 ± 0.1323**	**0.3944 ± 0.3059**
	nnPU	0.3703 ± 0.0421	**0.3116 ± 0.1362**	0.4563 ± 0.3314	**0.4229 ± 0.0336**	0.3333 ± 0.1255	**0.3299 ± 0.1070**
MLP	uPU	0.4045 ± 0.0339	0.3547 ± 0.0924	0.3744 ± 0.1555	**0.4386 ± 0.0291**	**0.3193 ± 0.1028**	**0.3176 ± 0.0894**
	nnPU	0.3525 ± 0.0441	**0.2697 ± 0.1749**	0.4194 ± 0.3296	**0.4504 ± 0.0425**	0.3486 ± 0.1056	**0.3334 ± 0.1400**
LSTM	uPU	0.4571 ± 0.0442	**0.3305 ± 0.1092**	0.3220 ± 0.1143	**0.5056 ± 0.0352**	0.3521 ± 0.0792	**0.2973 ± 0.1253**
	nnPU	0.4403 ± 0.0512	**0.3505 ± 0.1662**	0.4438 ± 0.3166	**0.4746 ± 0.0380**	0.3754 ± 0.1069	**0.3317 ± 0.1727**
NELA		Baseline F1-score	Baseline ΔDP	Baseline ΔEOd	Fairness-aware Learning F1-score	Fairness-aware Learning ΔDP	Fairness-aware Learning ΔEOd
Linear	uPU	0.7855 ± 0.0014	0.0042 ± 0.0029	0.0182 ± 0.0108	**0.7896 ± 0.0004**	**0.0014 ± 0.0008**	**0.0180 ± 0.0042**
	nnPU	0.7877 ± 0.0010	0.0086 ± 0.0104	0.0278 ± 0.0224	**0.7899 ± 0.0005**	**0.0018 ± 0.0013**	**0.0214 ± 0.0042**
MLP	uPU	0.7702 ± 0.0017	0.0915 ± 0.0071	0.0540 ± 0.0150	**0.7783 ± 0.0053**	**0.0376 ± 0.0372**	**0.0355 ± 0.0213**
	nnPU	0.7719 ± 0.0019	0.0890 ± 0.0070	0.0556 ± 0.0136	**0.7792 ± 0.0043**	**0.0334 ± 0.0348**	**0.0363 ± 0.0225**
LSTM	uPU	0.7622 ± 0.0134	0.1122 ± 0.0396	0.0605 ± 0.0310	**0.7863 ± 0.0021**	**0.0035 ± 0.0036**	**0.0103 ± 0.0085**
	nnPU	**0.7932 ± 0.0029**	0.1168 ± 0.0163	0.0560 ± 0.0123	0.7792 ± 0.0124	**0.0096 ± 0.0094**	**0.0263 ± 0.0212**

Table 2: Experimental results for **online** learning with and without fairness constraints. The superior results (higher F1-score; lower ΔDP and ΔEOd) for each evaluation metric are **bolded** for each combination of model, PU method, and dataset, comparing the baseline without fairness constraints to the model with fairness constraints.

$1, 2, \cdots, T$) with subset size b where T is the number of total training rounds. At t-th training round, $f_t = f(\boldsymbol{x}_t, \boldsymbol{w}_t) = \sum_{i=1}^{b} \boldsymbol{w}_t^\top \cdot \boldsymbol{x}_t^{(i)}$ where f is a linear classifier, and $\boldsymbol{w}_t \in F$ is a learnable weight vector for a convex set F. By adding L_2 regularizer and a conservative constraint to the PU risk estimator, the final objective function of *fairness-aware online PU learning* (FOPU) becomes

$$R_{I_t}(f_t) = R_{\text{pu}}(f_t) + \lambda_r \Omega(f_t) + \lambda_f R_{\text{fair}}(f_t) + \frac{\gamma_t}{2} \|\boldsymbol{w}_t - \boldsymbol{w}_{t-1}\|_2^2 \tag{4}$$

where γ, λ_r, and λ_f are hyperparemeters, and $\Omega(f_t) = \frac{\|\boldsymbol{w}_t\|_2^2}{2}$ is a parameter regularizer. We set $\gamma_t = \gamma + \lambda_r t$ with $\gamma = 1/\sqrt{b}$ as suggested in (Zhang et al., 2021). The last term limits the drastic changes of the weight to avoid overfitting to newly provided data. More details about optimization for online learning is introduced in Appendix C.

4 Theoretical Analysis

In the previous literature, the fairness violation in online learning has not been studied. Although (Zhao et al., 2021) shows a $\mathcal{O}(\sqrt{T \log T})$ bound of long-term fairness constraint, it is limited to the online meta-learning and not applicable to real-time online learning like FOPU. Furthermore, the impact of online learning with neural networks on fairness has not been studied either at each round. We prove that the cumulative fairness regret bound

Wiki		Baseline (Offline)			Fairness-aware Learning (Offline)		
		F1-score	ΔDP	ΔEOd	F1-score	ΔDP	ΔEOd
BERT	uPU	**0.6987±0.0055**	0.2281±0.0154	0.0992±0.0129	0.6733±0.0199	**0.1931±0.0319**	**0.0882±0.0308**
	nnPU	0.7091±0.0073	0.2326±0.0137	0.0819±0.0170	**0.7132±0.0059**	**0.2215±0.0091**	**0.0774±0.0138**
Distill	uPU	0.7114±0.0020	0.2496±0.0060	0.1217±0.0078	**0.7155±0.0048**	**0.2126±0.0078**	**0.0506±0.0138**
	nnPU	**0.7374±0.0038**	0.2400±0.0189	0.1159±0.0293	0.7346±0.0013	**0.2026±0.0098**	**0.0384±0.0111**
Chat Toxicity		Baseline (Offline)			Fairness-aware Learning (Offline)		
		F1-score	ΔDP	ΔEOd	F1-score	ΔDP	ΔEOd
BERT	uPU	**0.5603±0.0182**	0.5010±0.0571	0.5623±0.0750	0.5437±0.0116	**0.4247±0.0995**	**0.4382±0.1471**
	nnPU	**0.5860±0.0200**	0.4626±0.0915	0.4645±0.1660	0.5759±0.0142	**0.3809±0.0960**	**0.3370±0.1329**
Distill	uPU	**0.5941±0.0211**	0.4905±0.0995	0.4813±0.1665	0.5929±0.0189	**0.4503±0.0921**	**0.4241±0.1453**
	nnPU	0.6007±0.0133	0.4792±0.1041	0.4462±0.1804	**0.6009±0.0183**	**0.4723±0.1060**	**0.4331±0.2048**
NELA		Baseline (Offline)			Fairness-aware Learning (Offline)		
		F1-score	ΔDP	ΔEOd	F1-score	ΔDP	ΔEOd
BERT	uPU	0.8202±0.0021	0.1950±0.0091	0.0468±0.0144	**0.8245±0.0017**	**0.1865±0.0141**	**0.0312±0.0153**
	nnPU	0.8174±0.0029	**0.1670±0.0120**	0.0533±0.0175	**0.8227±0.0027**	0.2100±0.0074	**0.0275±0.0107**
Distill	uPU	0.8289±0.0019	**0.1804±0.0061**	0.0378±0.0111	**0.8325±0.0022**	0.1935±0.0178	**0.0248±0.0116**
	nnPU	0.8303±0.0019	**0.1891±0.0109**	0.0213±0.0124	**0.8309±0.0017**	0.1953±0.0117	**0.0129±0.0077**
Wiki		Baseline (Online)			Fairness-aware Learning (Online)		
		F1-score	ΔDP	ΔEOd	F1-score	ΔDP	ΔEOd
BERT	uPU	**0.6953±0.0022**	0.1903±0.0104	0.0971±0.0062	0.6881±0.0019	**0.1790±0.0059**	**0.0844±0.0094**
	nnPU	**0.6905±0.0022**	0.1862±0.0081	0.0877±0.0116	0.6822±0.0022	**0.1755±0.0078**	**0.0830±0.0098**
Distill	uPU	**0.6966±0.0020**	0.2412±0.0057	0.1202±0.0095	0.6861±0.0016	**0.2044±0.0042**	**0.0674±0.0076**
	nnPU	**0.6902±0.0030**	0.2343±0.0064	0.1063±0.0083	0.6790±0.0019	**0.2083±0.0074**	**0.0688±0.0096**
Chat Toxicity		Baseline (Online)			Fairness-aware Learning (Online)		
		F1-score	ΔDP	ΔEOd	F1-score	ΔDP	ΔEOd
BERT	uPU	**0.4891±0.0308**	0.4427±0.0536	0.5169±0.1171	0.4753±0.0563	**0.4176±0.0776**	**0.4754±0.1574**
	nnPU	0.4875±0.0275	0.4660±0.0781	0.5492±0.1463	**0.4918±0.0363**	**0.4373±0.0908**	**0.4938±0.1744**
Distill	uPU	**0.5107±0.0343**	0.4806±0.0609	0.5381±0.1285	0.5010±0.0250	**0.4040±0.0701**	**0.4562±0.1148**
	nnPU	**0.5169±0.0359**	0.4750±0.0834	0.5291±0.1449	0.5116±0.0370	**0.4254±0.0857**	**0.4577±0.1507**
NELA		Baseline (Online)			Fairness-aware Learning (Online)		
		F1-score	ΔDP	ΔEOd	F1-score	ΔDP	ΔEOd
BERT	uPU	**0.7983±0.0009**	**0.1027±0.0072**	0.0770±0.0088	0.7978±0.0012	0.1309±0.0100	**0.0419±0.0143**
	nnPU	**0.8161±0.0015**	**0.1448±0.0178**	0.0519±0.0203	0.8160±0.0019	0.1485±0.0143	**0.0440±0.0148**
Distill	uPU	**0.8034±0.0009**	0.1219±0.0113	0.0601±0.0251	**0.8034±0.0008**	**0.1075±0.0157**	**0.0395±0.0298**
	nnPU	**0.8035±0.0012**	**0.1113±0.0195**	0.0456±0.0291	0.8034±0.0013	0.1134±0.0184	**0.0328±0.0236**

Table 3: Experimental results for **offline** and **online** learning with and without fairness constraints for pre-trained language model, BERT and DistillBERT. The superior results (higher F1-score; lower ΔDP and ΔEOd) for each evaluation metric are **bolded** for each combination of model, PU method, and dataset, comparing the baseline without fairness constraints to the model with fairness constraints.

in OGD such that $\mathcal{O}(\frac{\sqrt{T}}{b})$ where b is the size of incoming dataset. It indicates online learning framework with a linear classifier affects the fairness violation in two ways, the total number of round T and the size of incoming data b. In the special case of online learning such that only a single datum is provided at each round, this proof still holds with a single batch size, $b = 1$. Moreover, we show the usage of MLP in online PU learning also affects the fairness regret compared to a linear classifier, making $\mathcal{O}(\sqrt{T \log L} + \frac{\sqrt{T}}{b})$ bound where L is the number of layer in MLP. All assumptions and proofs are elaborated in Appendix E.

5 Experimental Results

5.1 Implementation Detail

In this paper, we utilize three different NLP datasets: Wikipedia Talk (Thain et al., 2017; Wulczyn et al., 2017) and Chat Toxicity (Lin et al., 2023) datasets for toxicity classification, and NELA-2018 dataset (Nørregaard et al., 2019) for misinformation detection. Toxicity classification is prone to bias, particularly as documents containing sexuality-related terms are often misclassified as toxic, resulting in an increased false positive rate. For the NELA-2018 dataset (Nørregaard et al., 2019), the sensitive attribute raising fairness concerns is the political leaning, either left or right, as indicated in (Park et al., 2022). All datasets are divided into 60%, 20%, and 20% splits for training, validation, and testing, respectively.

As only positive-negative labels are given in the dataset, we replace them with positive-unlabeled settings using a hyperparameter, unlabeled positive ratio γ_u, indicating the portion of positive samples turned into unlabeled along with all the negative samples. For example, when $\gamma_u = 0.4$, 40% of positive samples and all negative samples are regarded as unlabeled. We employ $\gamma_u = 0.5$ to report performance in Tables 1 and 2, while the impact of γ_u and the robustness of FOPU against γ_u are discussed in Fig.3.

We conduct extensive experiments to validate the feasibility of our proposed Fairness-Aware Online PU learning as well as offline learning. Two different PU approaches, uPU and nnPU are implemented for three different classifiers, linear, MLP,

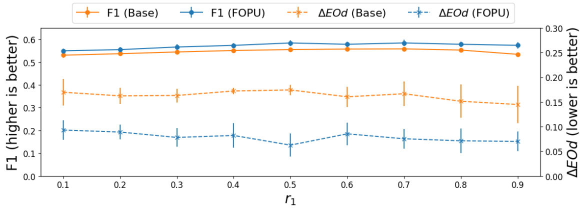

Figure 3: The experimental results with online MLP and nnPU for Wiki dataset varying γ_u show that the fairness constraint consistently improves fairness by lowering ΔEOd while preserving F1 score.

and LSTM. In the online setting, we conduct extensive experiments with the fixed total number of rounds $T = 200$, where only $b = N/T$ samples are provided at each round only once, where N is the total number of training samples. More details of implementation are introduced in Appendix G.

5.2 Result Analysis

We successfully integrate fairness constraints, PU learning, and online learning for all classifiers. As shown in Tables 1 and 2, the fairness constraint, Eq.(3), effectively improves targeted fairness metric, ΔEOd, while maintaining comparable F1-scores across all datasets and PU baselines. Additionally, Fig.3 shows that applying the fairness constraint in an online learning setting consistently enhances fairness for all γ_u values, while preserving F1-scores comparable to the baseline.

5.3 Extension to Pre-trained Language Models

With the growing adaptability of pre-trained language models, our approach can be effectively extended to such models, followed by a linear classifier. Specifically, instead of utilizing Doc2Vec (Le and Mikolov, 2014) for vectorization in linear, MLP, or LSTM classifiers, we leverage pre-trained models like BERT (Devlin, 2018) and DistilBERT (Sanh, 2019) as feature extractors, with a linear classifier applied on the representations. Since the primary objective is fair classification, training only the final linear classifier has been demonstrated as an efficient strategy to obtain fair prediction, as evidenced in (Mao et al., 2023).

In our experiments applying FOPU to pre-trained BERT and DistilBERT models, our framework effectively reduces the ΔEOd while preserving the F1 score, as shown in the Table 3. The results underscore the flexibility of our method in integrating with pre-trained language models while retaining a strong theoretical basis by restricting training to the linear classifier alone.

5.4 Limitation

We have considered recent PU learning methods such as Dist-PU (Zhao et al., 2022) and Robust-PU (Zhu et al., 2023). However, these approaches require a significant number of data points during training, making them more suitable for static settings. For example, Dist-PU compares the label distribution of predicted results with ground truth, requiring a large dataset to accurately align the distributions. In an online setting, where only limited data is available at each iteration, the label distribution in the prediction set may become skewed, restricting the applicability of Dist-PU. Similarly, Robust-PU iteratively refines the selection of negative samples from unlabeled data by adjusting hardness thresholds, which also necessitates a substantial number of unlabeled samples per iteration—an unrealistic requirement in an online context.

Given these constraints, we prioritize PU learning methods that rely solely on designing a risk estimator such as uPU and nnPU, which is more suited to online learning.

6 Conclusion

In this study, we address the issue of fairness in Positive-Unlabeled (PU) learning in text classification, particularly in the challenging context of online learning. We emphasize the necessity of strategies that ensure fairness in scenarios where data is incrementally provided, and only positive and unlabeled data are available. Our approach aims to enhance fairness in PU learning and extend it to online learning for both linear and deep neural network classifiers. We demonstrate that incorporating a convex fairness constraint during the training significantly improves fairness metrics (ΔEOd) while maintaining the F1-score. Additionally, we delve into the mathematical foundations of fairness in online settings by proving a cumulative fairness loss, i.e. fair regret bound.

Acknowledgements

This work was partially supported by the EMBRIO Institute, contract #2120200, a National Science Foundation (NSF) Biology Integration Institute, Purdue's Elmore ECE Emerging Frontiers Center, and NSF IIS #1955890, IIS #2146091, IIS #2345235.

References

Darko Androcec. 2020. Machine learning methods for toxic comment classification: a systematic review. *Acta Universitatis Sapientiae, Informatica*, 12(2):205–216.

Jessa Bekker and Jesse Davis. 2020. Learning from positive and unlabeled data: A survey. *Machine Learning*, 109:719–760.

Junyi Chai and Xiaoqian Wang. 2022. Fairness with adaptive weights. In *International Conference on Machine Learning*, pages 2853–2866. PMLR.

Marthinus Christoffel, Gang Niu, and Masashi Sugiyama. 2016. Class-prior estimation for learning from positive and unlabeled data. In *Asian Conference on Machine Learning*, pages 221–236. PMLR.

Mariana Caravanti de Souza, Bruno Magalhães Nogueira, Rafael Geraldeli Rossi, Ricardo Marcondes Marcacini, Brucce Neves Dos Santos, and Solange Oliveira Rezende. 2022. A network-based positive and unlabeled learning approach for fake news detection. *Machine learning*, 111(10):3549–3592.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Marthinus Du Plessis, Gang Niu, and Masashi Sugiyama. 2015. Convex formulation for learning from positive and unlabeled data. In *International conference on machine learning*, pages 1386–1394. PMLR.

Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220.

Yoav Freund and Robert E Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139.

Jin Ho Go, Alina Sari, Jiaojiao Jiang, Shuiqiao Yang, and Sanjay Jha. 2022. Fake news quick detection on dynamic heterogeneous information networks. *arXiv preprint arXiv:2205.07039*.

Alex Graves and Alex Graves. 2012. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.

Cho-Jui Hsieh, Nagarajan Natarajan, and Inderjit Dhillon. 2015. Pu learning for matrix completion. In *International conference on machine learning*, pages 2445–2453. PMLR.

Taeuk Jang, Pengyi Shi, and Xiaoqian Wang. 2022. Group-aware threshold adaptation for fair classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6988–6995.

Taeuk Jang, Feng Zheng, and Xiaoqian Wang. 2021. Constructing a fair classifier with generated fair data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7908–7916.

Masahiro Kato, Takeshi Teshima, and Junya Honda. 2019. Learning from positive and unlabeled data with a selection bias. In *International conference on learning representations*.

Joon Sik Kim, Jiahao Chen, and Ameet Talwalkar. 2020. Model-agnostic characterization of fairness tradeoffs. *arXiv preprint arXiv:2004.03424*.

Ryuichi Kiryo, Gang Niu, Marthinus C Du Plessis, and Masashi Sugiyama. 2017. Positive-unlabeled learning with non-negative risk estimator. *Advances in neural information processing systems*, 30.

Shuchen Kong, Weiwei Shen, Yingbin Zheng, Ao Zhang, Jian Pu, and Jun Wang. 2019. False positive rate control for positive unlabeled learning. *Neurocomputing*, 367:13–19.

Keita Kurita, Anna Belova, and Antonios Anastasopoulos. 2019. Towards robust toxic content classification. *arXiv preprint arXiv:1912.06872*.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR.

Changchun Li, Ximing Li, Lei Feng, and Jihong Ouyang. 2022a. Who is your right mixup partner in positive and unlabeled learning. In *International Conference on Learning Representations*.

Peizhao Li and Hongfu Liu. 2022. Achieving fairness at no utility cost via data reweighing with influence. In *International Conference on Machine Learning*, pages 12917–12930. PMLR.

Zhen Li, Xiting Wang, Weikai Yang, Jing Wu, Zhengyan Zhang, Zhiyuan Liu, Maosong Sun, Hui Zhang, and Shixia Liu. 2022b. A unified understanding of deep nlp models for text classification. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4980–4994.

Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. 2023. Toxicchat: Unveiling hidden challenges of toxicity detection in real-world user-ai conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4694–4702.

Yuzhen Mao, Zhun Deng, Huaxiu Yao, Ting Ye, Kenji Kawaguchi, and James Zou. 2023. Last-layer fairness fine-tuning is simple and effective for neural networks. *arXiv preprint arXiv:2304.03935*.

Jeppe Nørregaard, Benjamin D Horne, and Sibel Adalı. 2019. Nela-gt-2018: A large multi-labelled news dataset for the study of misinformation in news articles. In *Proceedings of the international AAAI conference on web and social media*, volume 13, pages 630–638.

Jinkyung Park, Rahul Ellezhuthil, Ramanathan Arunachalam, Lauren Feldman, and Vivek Singh. 2022. Toward fairness in misinformation detection algorithms. In *Workshop Proceedings of the 16th International AAAI Conference on Web and Social Media. Retrieved from https://doi. org/10.36190.*

Vishakha Patil, Ganesh Ghalme, Vineet Nair, and Yadati Narahari. 2021. Achieving fairness in the stochastic multi-armed bandit problem. *The Journal of Machine Learning Research*, 22(1):7885–7915.

Amirarsalan Rajabi and Ozlem Ozmen Garibay. 2022. Tabfairgan: Fair tabular data generation with generative adversarial networks. *Machine Learning and Knowledge Extraction*, 4(2):488–501.

Doyen Sahoo, Quang Pham, Jing Lu, and Steven CH Hoi. 2017. Online deep learning: Learning deep neural networks on the fly. *arXiv preprint arXiv:1711.03705*.

V Sanh. 2019. Distilbert, a distilled version of bert: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Nithum Thain, Lucas Dixon, and Ellery Wulczyn. 2017. Wikipedia Talk Labels: Toxicity.

Anuruddha Thennakoon, Chee Bhagyani, Sasitha Premadasa, Shalitha Mihiranga, and Nuwan Kuruwitaarachchi. 2019. Real-time credit card fraud detection using machine learning. In *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pages 488–493. IEEE.

Yongkai Wu, Lu Zhang, and Xintao Wu. 2019. On convexity and bounds of fairness-aware classification. In *The World Wide Web Conference*, pages 3356–3362.

Ziwei Wu and Jingrui He. 2022. Fairness-aware model-agnostic positive and unlabeled learning. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1698–1708.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex machina: Personal attacks seen at scale. In *Proceedings of the 26th international conference on world wide web*, pages 1391–1399.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rodriguez, and Krishna P Gummadi. 2017a. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, pages 1171–1180.

Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. 2017b. Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics*, pages 962–970. PMLR.

Chuang Zhang, Chen Gong, Tengfei Liu, Xun Lu, Weiqiang Wang, and Jian Yang. 2021. Online positive and unlabeled learning. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 2248–2254.

Chen Zhao, Feng Chen, and Bhavani Thuraisingham. 2021. Fairness-aware online meta-learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2294–2304.

Yunrui Zhao, Qianqian Xu, Yangbangyan Jiang, Peisong Wen, and Qingming Huang. 2022. Dist-pu: Positive-unlabeled learning from a label distribution perspective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14461–14470.

Zhangchi Zhu, Lu Wang, Pu Zhao, Chao Du, Wei Zhang, Hang Dong, Bo Qiao, Qingwei Lin, Saravan Rajmohan, and Dongmei Zhang. 2023. Robust positive-unlabeled learning via noise negative sample self-correction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3663–3673.

Martin Zinkevich. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 928–936.

A Investigating Separated Class Prior

As we extend the PU learning framework considering fairness with respect to the different demographic groups, the class priors for two sensitive groups might be different from each other. We re-formulate Eq.(1) by separating the risk estimator for two subgroups' sensitive information $a \in \{+1, -1\}$,

$$
\begin{aligned}
&R_{\text{upu}}(f) \\
&= \Big[\pi^{+}\mathbb{E}_p[\ell(f(\boldsymbol{X}^{+})] + [\mathbb{E}_u[\ell(-f(\boldsymbol{X}^{+}))] - \pi^{+}\mathbb{E}_p[\ell(-f(\boldsymbol{X}^{+}))]]\Big] \\
&+\Big[\pi^{-}\mathbb{E}_p[\ell(f(\boldsymbol{X}^{-})] + [\mathbb{E}_u[\ell(-f(\boldsymbol{X}^{-}))] - \pi^{-}\mathbb{E}_p[\ell(-f(\boldsymbol{X}^{-}))]]\Big]
\end{aligned}
$$

where the superscript indicates the sensitive groups such that $\pi^+ = p(y^+ = +1)$, $\pi^- = p(y^- = +1)$ with $(\boldsymbol{X}^+, y^+) \in \{(x, y)|x \in \boldsymbol{X}, y \in \boldsymbol{Y}, a = +1\}$, $(\boldsymbol{X}^-, y^-) \in \{(x, y)|x \in \boldsymbol{X}, y \in \boldsymbol{Y}, a = -1\}$. However, this method does not consistently mitigate bias arising from an imbalanced dataset since PU learning tends to assign positive labels to negative samples, even when class priors are correctly assigned for each demographic group. Based on this understanding, we recognize the need for a fairness constraint on PU learning and its impact.

B Fairness Constraint and Convexity

B.1 DP and EOd Constraints with Convexity

Optimizing fairness constraints is a popular in-processing approach in fairness-aware classification. Learning a fair classifier is formulated as optimizing the objective function with L_2 regularization ($\Omega(f)$) and fairness constraints such as the Difference of Demographic Parity (DP)

$$
\begin{aligned}
&\min_{f\in\mathcal{F}} R_{\text{pu}}(f) + \lambda_r\Omega(f) \\
&\text{subject to } |DP(f)| \le \tau
\end{aligned}
\tag{5}
$$

where f denotes the real-valued classifier with learnable parameter $\boldsymbol{w} \in \mathbb{R}^d$, $\Omega(f) = \frac{\|\boldsymbol{w}\|_2^2}{2}$, and λ_r is a hyperparameter. DP requires independence between the predicted outcome and the sensitive information $a \in \{+1, -1\}$, $P(\hat{y}|a = -1) = P(\hat{y}|a = +1)$, i.e. $\hat{y} \perp\!\!\!\perp a$. The empirical DP is

$$
DP(f) = \mathbb{E}\big[\frac{\mathbb{I}_{a=1}}{p_1}\mathbb{I}_{f(x)>0} - (1 - \frac{\mathbb{I}_{a=-1}}{1-p_1}\mathbb{I}_{f(x)<0})\big] \tag{6}
$$

where $p_1 = p(a = +1)$.

However, the linear fairness constraint in Eq.(5)-(6) is not suitable for online PU learning since the online framework requires the objective function to be convex (Zinkevich, 2003). Thus, we adopt a convex fairness constraint (Wu et al., 2019) based on relaxed form of Eq.(6) by replacing the indicator function to real-valued function f, and wrapping them in convex-concave surrogate function κ and δ to make the fairness constraint bounded by the lower and upper bound, so that to be convex.

$$
DP_\kappa(f) = \mathbb{E}\big[\tfrac{\mathbb{I}_{a=1}}{p_1}\kappa(f(x)) - \big(1 - \tfrac{\mathbb{I}_{a=-1}}{1-p_1}\kappa(-f(x))\big)\big]
$$

$$
DP_\delta(f) = \mathbb{E}\big[\tfrac{\mathbb{I}_{a=1}}{p_1}\delta(f(x)) - \big(1 - \tfrac{\mathbb{I}_{a=-1}}{1-p_1}\delta(-f(x))\big)\big]
$$

where κ is a convex surrogate function $\kappa(z) = \max(z+1, 0)$ and δ is a concave surrogate function $\delta(z) = \min(z, 1)$ as proposed in (Wu et al., 2019). Therefore, optimizing the fairness constraint in Eq.(5) becomes a convex problem

$$
\begin{aligned}
&\min_{f\in\mathcal{F}} R_{\text{pu}}(f) + \lambda_r\Omega(f) \\
&\text{subject to } DP_\kappa(f) \le \tau, \\
&\text{subject to } -DP_\delta(f) \le \tau.
\end{aligned}
$$

However, the usefulness of DP is limited to cases where there exists a correlation between y and a such that $y \not\perp\!\!\!\perp a$. Difference of Equalized Odds (EOd) overcomes the limitation of DP by conditioning the metric on the ground truth Y, i.e. $P(\hat{y}|a = +1, y) = P(\hat{y}|a = -1, y), \forall y \in \{+1, -1\}$. Define $\pi = p(y = +1)$, $p(y = -1) = 1 - \pi$, $p_{1,1} = P(a = +1, y = +1)$ and $p_{1,-1} = P(a = +1, y = -1)$, EOd can be rewritten as,

$$
\begin{aligned}
&EOd(f) \\
&= \mathbb{E}\Big[\frac{\mathbb{I}_{a=1,y=1}}{p_{1,1}}\mathbb{I}_{f(x)>0} - \big(1 - \frac{\mathbb{I}_{a=-1,y=1}}{\pi - p_{1,1}}\mathbb{I}_{f(x)<0}\big)\Big] \\
&+ \mathbb{E}\Big[\tfrac{\mathbb{I}_{a=1,y=-1}}{p_{1,-1}}\mathbb{I}_{f(x)>0} - \big(1 - \tfrac{\mathbb{I}_{a=-1,y=-1}}{1-\pi-p_{1,-1}}\mathbb{I}_{f(x)<0}\big)\Big]
\end{aligned}
\tag{7}
$$

We extend the fairness constraint by deriving a convex form of EOd,

$$
\begin{aligned}
EOd_\kappa(f) &= \mathbb{E}\Big[\tfrac{\mathbb{I}_{a=1,y=1}}{p_{1,1}}\kappa(f(x)) - \big(1 - \tfrac{\mathbb{I}_{a=-1,y=1}}{\pi-p_{1,1}}\kappa(-f(x))\big)\Big] \\
&+\mathbb{E}\Big[\tfrac{\mathbb{I}_{a=1,y=-1}}{p_{1,-1}}\kappa(f(x)) - \Big(1 - \tfrac{\mathbb{I}_{a=-1,y=-1}}{1-\pi-p_{1,-1}}\kappa(-f(x))\Big)\Big]
\end{aligned}
\tag{8}
$$

$$
\begin{aligned}
EOd_\delta(f) &= \mathbb{E}\Big[\tfrac{\mathbb{I}_{a=1,y=1}}{p_{1,1}}\delta(f(x)) - \big(1 - \tfrac{\mathbb{I}_{a=-1,y=1}}{\pi-p_{1,1}}\delta(-f(x))\big)\Big] \\
&+\mathbb{E}\Big[\tfrac{\mathbb{I}_{a=1,y=-1}}{p_{1,-1}}\delta(f(x)) - \big(1 - \tfrac{\mathbb{I}_{a=-1,y=-1}}{1-\pi-p_{1,-1}}\delta(-f(x))\big)\Big]
\end{aligned}
\tag{9}
$$

DP and EOd will be used as evaluation metrics to verify each model's performance, while their convex form is used as a part of the objective function. The detailed derivation for EOd is introduced in next section.

B.2 Details of the convex form of Equalized Odds (EOd) constraint

From the definition of DP, we can obtain a similar expression for EOd by conditioning DO for each $y \in \{+1, -1\}$. The Difference of Equalized Odds (EOd) is

$$EOd(f) = \Big[\frac{1}{|\mathbb{I}_{a=1,y=1}|}\sum_{S_{+1,+1}} \mathbb{I}_{f(x)>0} - \frac{1}{|\mathbb{I}_{a=-1,y=1}|}\sum_{S_{-1,1}} \mathbb{I}_{f(x)>0}\Big] + \Big[\frac{1}{|\mathbb{I}_{a=1,y=-1}|}\sum_{S_{+1,-1}} \mathbb{I}_{f(x)>0} - \frac{1}{|\mathbb{I}_{a=-1,y=-1}|}\sum_{S_{-1,-1}} \mathbb{I}_{f(x)>0}\Big],$$

where $S_{a,y}, a \in \{+1, -1\}, y \in \{+1, -1\}$ is a subgroup with corresponding a and y. and can be rewritten in the expected form as

$$EOd(f) = \mathbb{E}\Big[\frac{\mathbb{I}_{a=1,y=1}}{p_{1,1}}\mathbb{I}_{f(x)>0} - \Big(1 - \frac{\mathbb{I}_{a=-1,y=1}}{\pi - p_{1,1}}\mathbb{I}_{f(x)<0}\Big)\Big] + \mathbb{E}\Big[\frac{\mathbb{I}_{a=1,y=-1}}{p_{1,-1}}\mathbb{I}_{f(x)>0} - \Big(1 - \frac{\mathbb{I}_{a=-1,y=-1}}{1-\pi-p_{1,-1}}\mathbb{I}_{f(x)<0}\Big)\Big],$$

since

$$1 = \mathbb{E}[\frac{\mathbb{I}_{a=-1,y=1}}{p_{-1,1}}] = \mathbb{E}[\frac{\mathbb{I}_{a=-1,y=1}}{\pi - p_{1,1}}] = \mathbb{E}[\frac{\mathbb{I}_{a=-1,y=1}}{\pi - p_{1,1}}\mathbb{I}_{f(x)<0} + \frac{\mathbb{I}_{a=-1,y=1}}{\pi - p_{1,1}}\mathbb{I}_{f(x)>0}],$$

$$1 = \mathbb{E}[\frac{\mathbb{I}_{a=-1,y=-1}}{p_{-1,-1}}] = \mathbb{E}[\frac{\mathbb{I}_{a=-1,y=-1}}{1 - \pi - p_{1,-1}}] = \mathbb{E}[\frac{\mathbb{I}_{a=-1,y=-1}}{1-\pi-p_{1,-1}}\mathbb{I}_{f(x)<0} + \frac{\mathbb{I}_{a=-1,y=-1}}{1-\pi-p_{1,-1}}\mathbb{I}_{f(x)>0}]$$

where $\pi = p(y = 1), p(y = -1) = 1 - \pi, p_{1,1} = P(a = 1, y = 1)$ and $p_{1,-1} = P(a = 1, y = -1)$.

EOd can be expressed as a convex form,

$$EOd_\kappa(f) = \mathbb{E}\Big[\frac{\mathbb{I}_{a=1,y=1}}{p_{1,1}}\kappa(f(x)) - \Big(1 - \frac{\mathbb{I}_{a=-1,y=1}}{\pi - p_{1,1}}\kappa(-f(x))\Big)\Big] + \mathbb{E}\Big[\frac{\mathbb{I}_{a=1,y=-1}}{p_{1,-1}}\kappa(f(x)) - \Big(1 - \frac{\mathbb{I}_{a=-1,y=-1}}{1-\pi-p_{1,-1}}\kappa(-f(x))\Big)\Big]$$

$$EOd_\delta(f) = \mathbb{E}\Big[\frac{\mathbb{I}_{a=1,y=1}}{p_{1,1}}\delta(f(x)) - \Big(1 - \frac{\mathbb{I}_{a=-1,y=1}}{\pi - p_{1,1}}\delta(-f(x))\Big)\Big] + \mathbb{E}\Big[\frac{\mathbb{I}_{a=1,y=-1}}{p_{1,-1}}\delta(f(x)) - \Big(1 - \frac{\mathbb{I}_{a=-1,y=-1}}{1-\pi-p_{1,-1}}\delta(-f(x))\Big)\Big]$$

If we replace the target fairness constraint to EOd rather than DP, the convex form of fairness constraint in objective function R_{EOd} is defined

$$R_{\text{EOd}}(f) = \begin{cases} EOd_\kappa(f) & \text{if } EOd(f) \geq 0 \\ EOd_\delta(f) & \text{if } EOd(f) < 0. \end{cases}$$

B.3 Positive Rate Penalty

The current fairness constraint aims to minimize $(TPR_1 - TPR_0) + (FPR_1 - FPR_0)$, as outlined in Eq.(7)-(9). Although minimizing the overall EOd constraint can enhance fairness by reducing differences in TPR and FPR across groups, it carries the potential risk of lowering the TPR value. In tasks such as toxicity classification or misinformation detection, TPR (recall) is a critical metric (Kurita et al., 2019), and any reduction is undesirable. Despite adopting the risk estimator in PU learning to improve agreement between predictions and ground truth, it may not adequately prevent a TPR decrease when the number of positive instances is limited (e.g., 9.66% in the Wiki Toxicity dataset). Consequently, an additional constraint is necessary to avoid a decrease in TPR and an increase in FPR. This new constraint would penalize the model if the current TPR is lower or the FPR is higher than in the previous step. Furthermore, because the indicator function used in TPR and FPR calculations is not differentiable, we apply the sigmoid function in place of the indicator function, e.g., $TPR_1 = \frac{\sum_{a=1,y=1}\sigma(\hat{y})}{n_{11}}$.

$$\begin{aligned}\mathcal{L}_{\text{p}} = &\max(0, TPR_1^{base} - TPR_1^{(t)}) \\ &+ \max(0, TPR_0^{base} - TPR_0^{(t)}) \\ &+ \max(FPR_1^{(t)} - FPR_1^{base}, 0) \\ &+ \max(FPR_0^{(t)} - FPR_0^{base}, 0)\end{aligned} \quad (10)$$

where $TPR^{base} \leftarrow \max(TPR^{base}, TPR^{(t)})$ and $FPR^{base} \leftarrow \min(FPR^{base}, FPR^{(t)})$. Therefore, $R_{\text{fair}} \leftarrow R_{\text{fair}} + \mathcal{L}_{\text{p}}$.

B.4 Impact of Positive Rate Penalty

As discussed in the Section 3.3 and Appendix B.3, we employ a positive rate penalty term to mitigate the reduction of TPR when applying a fairness constraint. To verify its impact, we conducted an ablation study on the Wiki dataset, comparing the results of the fairness constraint with and without the positive rate penalty. Table 4 demonstrates that the positive rate penalty term significantly improves recall without compromising the fairness level.

C Online Learning Schemes

The weight vector $\boldsymbol{w}_t$ of the linear classifier f_t in Eq.(4) is updated by Online Gradient Descent

Wiki-Offline		W/O Positive Penalty			W/ Positive Penalty		
		F1	Recall	ΔEOd	F1	Recall	ΔEOd
Linear	uPU	0.5610 ± 0.0038	0.5727 ± 0.0130	**0.1607 ± 0.0156**	**0.5622 ± 0.0038**	**0.5792 ± 0.0123**	0.1620 ± 0.0175
	nnPU	0.5600 ± 0.0038	0.5704 ± 0.0132	**0.1575 ± 0.0113**	**0.5609 ± 0.0035**	**0.5763 ± 0.0120**	**0.1575 ± 0.0128**
MLP	uPU	0.5931 ± 0.0096	0.6737 ± 0.0547	**0.0550 ± 0.0163**	**0.6033 ± 0.0094**	**0.7386 ± 0.0217**	0.0798 ± 0.0324
	nnPU	0.5604 ± 0.0050	0.6493 ± 0.0296	**0.0551 ± 0.0223**	**0.5849 ± 0.0105**	**0.7779 ± 0.0210**	0.0589 ± 0.0155
LSTM	uPU	0.5894 ± 0.0189	0.5007 ± 0.0363	**0.0396 ± 0.0191**	**0.6216 ± 0.0097**	**0.5959 ± 0.0311**	0.0558 ± 0.0191
	nnPU	0.6407 ± 0.0069	0.6479 ± 0.0274	**0.0352 ± 0.0227**	**0.6433 ± 0.0056**	**0.6638 ± 0.0327**	0.0382 ± 0.0204
Wiki-Online		**W/O Positive Penalty**			**W/ Positive Penalty**		
		F1	Recall	ΔEOd	F1	Recall	ΔEOd
Linear	uPU	0.5593 ± 0.0028	0.5704 ± 0.0157	**0.1475 ± 0.0172**	**0.5601 ± 0.0026**	**0.5722 ± 0.0110**	0.1506 ± 0.0209
	nnPU	0.5597 ± 0.0027	0.5874 ± 0.0182	**0.1490 ± 0.0148**	**0.5633 ± 0.0020**	**0.5999 ± 0.0182**	0.1531 ± 0.0221
MLP	uPU	0.5519 ± 0.0069	0.5920 ± 0.0166	0.1601 ± 0.0307	**0.5544 ± 0.0076**	**0.6450 ± 0.0309**	**0.1505 ± 0.0202**
	nnPU	0.5505 ± 0.0078	0.5793 ± 0.0208	0.1516 ± 0.0341	**0.5545 ± 0.0073**	**0.6433 ± 0.0191**	**0.1463 ± 0.0201**
LSTM	uPU	0.5546 ± 0.0152	0.6159 ± 0.0892	**0.1098 ± 0.0244**	**0.5583 ± 0.0080**	**0.6215 ± 0.0639**	0.1107 ± 0.0247
	nnPU	**0.5545 ± 0.0176**	0.6712 ± 0.0852	**0.1149 ± 0.0248**	0.5507 ± 0.0178	**0.6950 ± 0.0667**	0.1168 ± 0.0252

Table 4: Ablation study on the effect of the positive rate penalty term within the fairness constraint.

(OGD) (Zinkevich, 2003) at t-th time step,

$$\boldsymbol{w}_t \leftarrow \Pi_{\mathcal{W}}(\boldsymbol{w}_{t-1} - \eta_t \nabla_t)$$

where $\eta_t = b/(\beta\sqrt{t})$ is a step size, $\beta = b/\eta_1$, and η_1 is the initial learning rate. ∇_t is the gradient of $R_{I_t}(f_t)$, and $\Pi_{\mathcal{W}}(\boldsymbol{w})$ is a projection step defined as $\Pi_{\mathcal{W}}(\boldsymbol{w}) = \arg\min_{\boldsymbol{w}' \in \mathcal{W}} \|\boldsymbol{w} - \boldsymbol{w}'\|$ with $\mathcal{W}$ being a feasible set of $\boldsymbol{w}$.

As OGD is designed only for linear classifiers, we further extend the framework for MLP using Online Deep Learning (ODL) (Sahoo et al., 2017). In (Sahoo et al., 2017), MLP is regarded as a mixture of experts considering each linear layer as an expert. The intermediate predictions are aggregated for the final prediction, and back-propagated by Hedge Backpropagation (Freund and Schapire, 1997). Since the deep neural networks for online PU learning have not been studied yet in previous literature, we modify the ODL framework to facilitate online PU learning with an MLP classifier, and apply ODL to LSTM. Details in Online Deep Learning are introduced in Appendix D.

D Online Deep Learning with Hedge Backpropagation

In this appendix, we elucidate our online deep learning framework which integrates the Hedge Backpropagation methodology. Traditional online learning models have been primarily constructed for linear models. When applied to Deep Neural Networks (DNNs), these conventional models face convergence difficulties, the notorious vanishing gradient problem, and challenges in determining an optimal network depth.

For a standard representation of a DNN, the relationship is defined as

$$\mathbf{F}(\mathbf{x}) = \mathrm{softmax}(W^{(L+1)}\mathbf{h}^{(L)}),$$

$$\mathbf{h}^{(l)} = \sigma(W^{(l)}\mathbf{h}^{(l-1)})$$

for all $l = 1, \cdots, L$, where $\mathbf{h}^{(0)} = \mathbf{x}$. In the Online Gradient Descent (OGD), the updating rule is expressed as

$$W_{t+1}^{(l)} \leftarrow W_t^{(l)} - \eta \nabla_{W_t^{(l)}} \mathcal{L}(\mathbf{F}(\mathbf{x}_t), y_t).$$

In the proposed Hedge Backpropagation, the network's prediction is a weighted sum of predictions from all layers:

$$\mathbf{F}(\mathbf{x}) = \sum_{l=0}^{L} \alpha^{(l)} \mathbf{f}^{(l)},$$
$$\mathbf{f}^{(l)} = \mathrm{softmax}(\mathbf{h}^{(l)}\Theta^{(l)}), \quad \forall l = 0, \cdots, L,$$
$$\mathbf{h}^{(l)} = \sigma(W^{(l)}\mathbf{h}^{(l-1)}), \quad \forall l = 1, \cdots, L.$$

New parameters $\Theta^{(l)}$ and $\alpha^{(l)}$ are introduced, where $\Theta^{(l)}$ is associated with each layer's output and $\alpha^{(l)}$ serves as a weight for all outputs across layers. The overall loss function is then formulated as

$$\mathcal{L}(\mathbf{F}(\mathbf{x}), y) = \sum_{l=0}^{L} \alpha^{(l)} \mathcal{L}(\mathbf{f}^{(l)}, y).$$

For the updating algorithm, we start with $\alpha^{(l)} = \frac{1}{L+1}$ for all $l = 0, \cdots, L$. During each iteration, classifier $\mathbf{f}^{(l)}$ predicts $\hat{y}_t^{(l)}$ and updates $\alpha_{t+1}^{(l)}$ using

$$\alpha_{t+1}^{(l)} \leftarrow \alpha_t^{(l)} \beta^{\mathcal{L}(\mathbf{f}^{(l)}(\mathbf{x}), y)},$$

where $\beta \in (0, 1)$ is the discount rate. Finally, both Θ and W are updated through OGD as detailed in the equations provided.

E Theoretical Analysis

In this section, we aim to investigate how online learning and deep neural networks with fairness constraints affect the cumulative fairness regret compared to offline learning.

Theorem E.1. *Consider $f_t : \mathcal{X} \rightarrow \mathbb{R}$ is a real valued linear function with learnable parameter $\boldsymbol{w}_t$ at round $t \in \{1, \cdots, T\}$ in online learning. Let $R_{fair}(f_t(\boldsymbol{w}_t))$ be a convex approximation of fairness constraint at t-th time step as defined in Eq.(3). Let $\{I_t\}_{t=1}^T$ be the incoming training data at the t-th time step where its size is $b = |I_t| > 0$. Denote $g_t = \nabla R_{fair}(f_t(\boldsymbol{w}_t))$ for simplicity and assume that $\|g_t\| \leq G$, $\|\boldsymbol{w}_t - \boldsymbol{w}_*\|^2 \leq K^2$, with constants $K, G > 0$ where $\boldsymbol{w}_*$ is an optimal weight obtained by the offline learning. Define the fair regret as*

$$Regret_T\big(R_{fair}(f(\boldsymbol{w}))\big) = \sum_{t=1}^{T} \mathbb{E}[R_{fair}(f_t(\boldsymbol{w}_t)) - R_{fair}(f_t(\boldsymbol{w}_*))],$$

then we have the Fair Regret Bound as follows:

$$Regret_T^{OGD}\big(R_{fair}(f(\boldsymbol{w}))\big) \leq \big(\frac{\beta^2 F^2 + 2G^2}{2b\beta}\big)\sqrt{T}, \tag{11}$$

where $\beta = b/\eta_1$, where b is the size of incoming dataset and η_1 is the initial learning rate. In the special case of online learning such that only a single datum is provided at each round, this proof still holds with a single batch size, $b = 1$.

Insights from Theorem E.1. Theorem E.1 indicates online learning framework with a linear classifier affects the fairness violation in two ways, the total number of round T and the size of incoming data b.

Moreover, we show the usage of MLP in online PU learning also affects the fairness regret compared to a linear classifier.

Theorem E.2. *Let $\mathbf{F} : \mathcal{X} \rightarrow \mathbb{R}$ be an Online Deep Learning framework with Hedge Backpropagation, where the final prediction is a weighted sum of each layer in MLP, i.e. $\mathbf{F}(\boldsymbol{w}) = \sum_{l=0}^{L} \alpha^{(l)} \mathbf{f}(\boldsymbol{w}^{(l)})$ where $\mathbf{f}(\boldsymbol{w}^{(l)})$ is each layer in MLP, $\alpha^{(l)}$ is multiplicative weight of each layer, and L is the number of layers. The cumulative fairness regret against a linear classifier is bounded by*

$$Regret_T^{Hedge}\big(R_{fair}(\mathbf{F}(\boldsymbol{w}))\big) \leq \frac{k+1}{k}\sqrt{T\ln(L+1)} \tag{12}$$

where $k = \sqrt{\frac{\ln(L+1)}{T}}/\epsilon$, $\epsilon = \ln(1/\mu)$, and $\mu \in (0, 1)$ is a constant discount rate paramter of multiplicative weight. In this research, $\mu = 0.99$ following (Sahoo et al., 2017).

Insights from Theorem E.1 and E.2. In Online Deep Learning with Hedge Backpropagation, the Theorem E.2 presents the cumulative fairness violation against a single linear classifier. On the other hand, each linear expert has its own fairness regret bound against the parameter obtained by offline learning as shown in Theorem E.1. Therefore, the final fairness violation of Hedge is the additive of two regret bounds.

Corollary E.3. *In Online Deep Learning with Hedge, there exists loosely Fair Regret bound against an offline linear classifier. From Eq.(15) and Eq.(12),*

$$\begin{aligned} Regret_T^{ODL}\big(R_{fair}(\mathbf{F}(\boldsymbol{w}))\big) &\leq Regret_T^{OGD} + Regret_T^{Hedge} \\ &= \frac{k+1}{k}\sqrt{T\ln(L+1)} + \big(\frac{\lambda_r^2 K^2 + 2G^2}{2b\lambda_r}\big)\sqrt{T}. \end{aligned} \tag{13}$$

The proofs for Theorem E.1 and Theorem E.2 are explained in Appendix F.1 and F.2, respectively.

F Proofs

F.1 Proof of Theorem 5.1

Consider $f_t : \mathcal{X} \rightarrow \mathbb{R}$ is a real valued linear function with learnable parameter $\boldsymbol{w}_t$ at round $t \in \{1, \cdots, T\}$ in online learning. Let $R_{\text{fair}}(f_t(\boldsymbol{w}_t))$ be a convex approximation of fairness constraint as an objective function at t-th time step. Let $\{I_t\}_{t=1}^T$ be the incoming training data at the t-th time step where its size is $b = |I_t| > 0$. Denote $g_t = \nabla R_{\text{fair}}(f_t(\boldsymbol{w}_t))$ for simplicity and assume that $\|g_t\| \leq G$, $\|\boldsymbol{w}_t - \boldsymbol{w}_*\|^2 \leq K^2$, with constants $K, G > 0$ where $\boldsymbol{w}_*$ is an optimal weight obtained by the offline learning. This assumption is valid since $\Pi_{\mathcal{W}}(\boldsymbol{w})$ is a projection step defined as $\Pi_{\mathcal{W}}(\boldsymbol{w}) = \arg\min_{\boldsymbol{w}' \in \mathcal{W}} \|\boldsymbol{w} - \boldsymbol{w}'\|$ with $\mathcal{W}$ being a feasible set of $\boldsymbol{w}$. Define the fair regret as

$$\text{Regret}_T\big(R_{\text{fair}}(f(\boldsymbol{w}))\big) = \sum_{t=1}^{T} \mathbb{E}[R_{\text{fair}}(f_t(\boldsymbol{w}_t)) - R_{\text{fair}}(f_t(\boldsymbol{w}_*))], \tag{14}$$

then we have the Fair Regret Bound as follows:

$$\text{Regret}_T^{\text{OGD}}\big(R_{\text{fair}}(f(\boldsymbol{w}))\big) \leq \big(\frac{\beta^2 K^2 + 2G^2}{2b\beta}\big)\sqrt{T}, \tag{15}$$

where $\beta = b/\eta_1$, where b is the size of incoming dataset and η_1 is the initial learning rate. In the special case of online learning such that only a single datum is provided at each round, this proof still holds with a single batch size, $b = 1$.

Proof. Let $\boldsymbol{w}_*$ be an optimal parameter obtained by the offline learning with the convex fairness constraint (3). As $R_{\text{fair}}(f_t(\boldsymbol{w}_t))$ is convex for all $\boldsymbol{w}$,

$$R_{\text{fair}}(f_t(\boldsymbol{w}_t)) \geq \nabla R_{\text{fair}}(f_t(\boldsymbol{w}_t))(\boldsymbol{w} - \boldsymbol{w}_t) + R_{\text{fair}}(f_t(\boldsymbol{w}_t))$$

From the definition of g_t,

$$\begin{aligned} & R_{\text{fair}}(f_t(\boldsymbol{w}_*)) \geq (\boldsymbol{w}_* - \boldsymbol{w}_t)g_t + R_{\text{fair}}(f_t(\boldsymbol{w}_t)) \\ \Leftrightarrow & R_{\text{fair}}(f_t(\boldsymbol{w}_t)) - R_{\text{fair}}(f_t(\boldsymbol{w}_*)) \leq (\boldsymbol{w}_t - \boldsymbol{w}_*)g_t \end{aligned} \tag{16}$$

The parameter $\boldsymbol{w}_t$ is updated by the Online Gradient Descent, $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \frac{\eta_t}{b}\sum_{i=1}^{b} g_{i,t}$ where η_t is a step size at the round t. Then,

$$\begin{aligned} (\boldsymbol{w}_{t+1} - \boldsymbol{w}_*)^2 &= (\boldsymbol{w}_t - \frac{\eta_t}{b}\sum_{i=1}^{b} g_{i,t} - \boldsymbol{w}_*)^2 \\ &= (\boldsymbol{w}_t - \boldsymbol{w}_*)^2 - \tfrac{2\eta_t}{b}(\boldsymbol{w}_t - \boldsymbol{w}_*)\textstyle\sum_{i=1}^{b} g_{i,t} + \tfrac{\eta_t^2}{b^2}\left\|\sum_{i=1}^{b} g_{i,t}\right\|^2 \\ &\leq (\boldsymbol{w}_t - \boldsymbol{w}_*)^2 - \frac{2\eta_t}{b}(\boldsymbol{w}_t - \boldsymbol{w}_*)\sum_{i=1}^{b} g_{i,t} + \frac{\eta_t^2}{b^2}G^2 \\ \Leftrightarrow & \frac{1}{b}(\boldsymbol{w}_t - \boldsymbol{w}_*)\sum_{i=1}^{b} g_{i,t} \\ &\leq \frac{1}{2\eta_t}\big((\boldsymbol{w}_t - \boldsymbol{w}_*)^2 - (\boldsymbol{w}_{t+1} - \boldsymbol{w}_*)^2\big) + \frac{\eta_t}{2b^2}G^2 \end{aligned} \tag{17}$$

From (14), (16) and (17),

$$\begin{aligned} & \text{Regret}_T^{\text{OGD}}\big(R_{\text{fair}}(f_t(\boldsymbol{w}_t))\big) \\ &= \sum_{t=1}^{T} \mathbb{E}[R_{\text{fair}}(f_t(\boldsymbol{w}_t)) - R_{\text{fair}}(f_t(\boldsymbol{w}_*))] \\ &\leq \sum_{t=1}^{T} \mathbb{E}[(\boldsymbol{w}_t - \boldsymbol{w}_*)g_t] \\ &= \sum_{t=1}^{T}(\frac{1}{b}(\boldsymbol{w}_t - \boldsymbol{w}_*)\sum_{i=1}^{b} g_{i,t}) \\ &\leq \frac{1}{2\eta_1}(\boldsymbol{w}_1 - \boldsymbol{w}_*)^2 - \frac{1}{2\eta_T}(\boldsymbol{w}_{T+1} - \boldsymbol{w}_*)^2 \\ &\quad + \frac{1}{2}\sum_{t=2}^{T}(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}})(\boldsymbol{w}_{t+1} - \boldsymbol{w}_*)^2 + \frac{G^2}{2b^2}\sum_{t=1}^{T}\eta_t \\ &\leq K^2\Big(\frac{1}{2\eta_1} + \frac{1}{2}\sum_{t=2}^{T}(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}})\Big) + \frac{G^2}{2b^2}\sum_{t=1}^{T}\eta_t \\ &\leq \frac{K^2}{2\eta_T} + \frac{G^2}{2b^2}\sum_{t=1}^{T}\eta_t \quad (\text{set } \eta_t = b/(\beta\sqrt{t})) \\ &= \frac{K^2}{2}\frac{\beta\sqrt{T}}{b} + \frac{G^2}{2b^2}\frac{b}{\beta}\sum_{t=1}^{T}\frac{1}{\sqrt{t}} \\ &\leq \frac{\beta K^2\sqrt{T}}{2b} + \frac{G^2}{2b\beta}\cdot 2\sqrt{T} \\ &= \frac{\beta K^2\sqrt{T}}{2b} + \frac{G^2\sqrt{T}}{b\beta} \\ &= (\frac{\beta^2K^2 + 2G^2}{2b\beta})\sqrt{T} \end{aligned} \tag{18}$$

□

F.2 Proof of Theorem 5.2

Let $\mathbf{F} : \mathcal{X} \to \mathbb{R}$ be an Online Deep Learning framework with Hedge Backpropagation, where the final prediction is a weighted sum of each layer in MLP, i.e. $\mathbf{F}(\boldsymbol{w}) = \sum_{l=0}^{L}\alpha^{(l)}\mathbf{f}(\boldsymbol{w}^{(l)})$ where $\mathbf{f}(\boldsymbol{w}^{(l)})$ is each layer in MLP, $\alpha^{(l)}$ is multiplicative weight of each layer, and L is the number of layers. The cumulative fairness regret against a single linear classifier (expert) is bounded by

$$\text{Regret}_T^{\text{Hedge}}\big(R_{\text{fair}}(\mathbf{F}(\boldsymbol{w}))\big) \leq \frac{k+1}{k}\sqrt{T\ln(L+1)} \tag{19}$$

where $k = \sqrt{\frac{\ln(L+1)}{T}}/\epsilon$, $\epsilon = \ln(1/\mu)$, and $\mu \in (0, 1)$ is a constant discount rate parameter of multiplicative weight. In this research, $\mu = 0.99$ following (Sahoo et al., 2017).

Proof. In Online Deep Learning, the final prediction is a weighted sum of each linear layer. At time step t,

$$\begin{aligned} \mathbf{F}_t(\boldsymbol{w}) &= \sum_{l=0}^{L}\alpha_t^{(l)}\mathbf{f}(\boldsymbol{w}_t^{(l)}) \\ \mathbf{f}(\boldsymbol{w}_t^{(l)}) &= \text{softmax}(\mathbf{h}_t^{(l)}\boldsymbol{w}_{t,out}^{(l)}), \forall l = 0, \cdots, L \\ \mathbf{h}_t^{(l)} &= \sigma(\boldsymbol{w}_{t,in}^{(l)}\mathbf{h}_t^{(l-1)}), \forall l = 1, \cdots, L \\ \mathbf{h}_t^{(0)} &= \mathbf{x}_t \end{aligned}$$

where $\boldsymbol{w}_{in}$ denotes the parameter between layers, and $\boldsymbol{w}_{out}$ is the parameter for computing each layer's output. $\alpha^{(l)}$ is a multiplicative weight across the all fairness cost R_{fair} of each layer, such that

$$R_{\text{fair}}(\mathbf{F}_t(\boldsymbol{w})) = \sum_{l=0}^{L}\alpha_t^{(l)}R_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l)})).$$

During the online training, $\boldsymbol{w}_{in}$ and $\boldsymbol{w}_{out}$ are updated by Online Gradient Descent by being regarded as an individual expert. The multiplicative weight is updated by

$$\alpha_{t+1}^{(l)} \leftarrow \alpha_t^{(l)} \mathrm{e}^{-\epsilon \mathrm{R_{fair}}(\mathbf{f}(\mathrm{w}_t^{(l)}))} \tag{20}$$

$$\alpha_{t+1}^{(l)} \leftarrow \frac{\alpha_{t+1}^{(l)}}{\sum_{l=0}^{L} \alpha_{t+1}^{(l)}}.$$

where we set $\alpha_1 = \frac{1}{1+L}$. Let $\epsilon > 0$ and all risk $R_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l)}))$ is non-negative. Set $\phi_t = \sum_{l=0}^{L} \alpha_t^{(l)}$ and $Z_t^{(l)} = \frac{\alpha_t^{(l)}}{\phi_t}$. The sum of multiplicative weights becomes

$$\phi_{t+1} = \sum_{l=0}^{L} \alpha_{t+1}^{(l)} = \sum_{l=0}^{L} \alpha_t^{(l)} \mathrm{e}^{-\epsilon \mathrm{R_{fair}}(\mathbf{f}(\mathrm{w}_t^{(l)}))}$$

$$= \phi_t \sum_{l=0}^{L} Z_t^{(l)} \mathrm{e}^{-\epsilon \mathrm{R_{fair}}(\mathbf{f}(\mathrm{w}_t^{(l)}))}$$

$$\leq \phi_t \sum_{l=0}^{L} Z_t^{(l)} \big(1 - \epsilon R_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l)})) + \epsilon^2 R_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l)}))^2\big)$$

$$(\because \mathrm{e}^{-\mathrm{x}} \leq 1 - \mathrm{x} + \mathrm{x}^2, \forall \mathrm{x} \geq 0)$$

$$= \phi_t \Big(1 - \epsilon \sum_{l=0}^{L} Z_t^{(l)} R_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l)})) + \epsilon^2 \sum_{l=0}^{L} Z_t^{(l)} R_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l)}))^2\Big)$$

$$(\because \sum_{l=0}^{L} Z_t^{(l)} = \sum_{l=0}^{L} \frac{\alpha_t^{(l)}}{\phi_t} = \frac{\phi_t}{\phi_t} = 1)$$

$$\leq \phi_t \exp\Big(-\epsilon \sum_{l=0}^{L} Z_t^{(l)} R_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l)})) + \epsilon^2 \sum_{l=0}^{L} Z_t^{(l)} R_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l)}))^2\Big)$$

$$(\because 1 + x \leq e^x)$$

$$= \phi_t \exp\Big(-\epsilon \boldsymbol{Z}_t \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t))) + \epsilon^2 \boldsymbol{Z}_t \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t)))^2\Big)$$

$$\Big(\because \text{denote} \sum_{l=0}^{L} Z_t^{(l)} R_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l)})) = \boldsymbol{Z}_t \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t))\Big)$$

Note that $\phi_1 = L + 1$ before the normalization and let $A_t = \exp\Big(-\epsilon \boldsymbol{Z}_t \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t))) + \epsilon^2 \boldsymbol{Z}_t\Big)$, then at the time step T,

$$\phi_T \leq \phi_{T-1} A_{t-1} \leq \phi_{T-2} A_{t-2} A_{t-1}$$

$$\leq \cdots \leq \phi_1 \Pi_{t=1}^{T-1} A_t \leq \phi_1 \Pi_{t=1}^{T} A_t \tag{21}$$

Then Eq.(21) becomes

$$\phi_T \leq (L+1) \exp\Big(-\epsilon \sum_{t=1}^{T} \boldsymbol{Z}_t \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t))) + \epsilon^2 \sum_{t=1}^{T} \boldsymbol{Z}_t \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t)))^2\Big).$$

For any expert l_*, by Eq.(20), the multiplicative weight at time T is $\alpha_T^{(l_*)} = \exp\Big(-\epsilon \sum_{t=1}^{T} \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l_*)}))\Big)$, while it is less than or equal to the sum of the weight, ϕ_T. Then,

$$\alpha_T^{(l_*)} = \exp\Big(-\epsilon \sum_{t=1}^{T} \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l_*)}))\Big) \leq \phi_T$$

$$\leq (L+1) \exp\Big(-\epsilon \sum_{t=1}^{T} \boldsymbol{Z}_t \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t))) + \epsilon^2 \sum_{t=1}^{T} \boldsymbol{Z}_t \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t)))^2\Big).$$

Taking the logarithm of both sides, we get

$$-\epsilon \sum_{t=1}^{T} \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l_*)}))$$

$$\leq \ln(L+1) - \epsilon \sum_{t=1}^{T} \boldsymbol{Z}_t \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t))) + \epsilon^2 \sum_{t=1}^{T} \boldsymbol{Z}_t \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t)))^2$$

Dividing by ϵ for both sides, we get

$$\sum_{t=1}^{T} \boldsymbol{Z}_t \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t))) - \sum_{t=1}^{T} \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l_*)}))$$

$$\leq \frac{\ln(L+1)}{\epsilon} + \epsilon \sum_{t=1}^{T} \boldsymbol{Z}_t \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t)))^2 \tag{22}$$

The left-hand side refers to the cumulative loss between Hedge and a single expert. In our fairness-aware training, $\boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l)})) \leq 1$ since it is a fairness measure. Then, (22) becomes

$$\text{Regret}_T^{Hedge}\big(R_{\text{fair}}(\mathbf{F}(\boldsymbol{w}))\big)$$

$$= \sum_{t=1}^{T} \boldsymbol{Z}_t \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t))) - \sum_{i=1}^{T} \boldsymbol{R}_{\text{fair}}(\mathbf{f}(\boldsymbol{w}_t^{(l_*)})$$

$$\leq \frac{\ln(L+1)}{\epsilon} + \epsilon \sum_{t=1}^{T} \boldsymbol{Z}_t$$

$$= \frac{\ln(L+1)}{\epsilon} + T\epsilon \quad \left(\text{set} \quad \epsilon = k\sqrt{\frac{\ln(L+1)}{T}}\right)$$
$$= \frac{k+1}{k}\sqrt{T\ln(L+1)} \tag{23}$$

□

G Implementation Details

In this paper, we utilize three different NLP datasets: Wikipedia Talk (Thain et al., 2017; Wulczyn et al., 2017) and Chat Toxicity (Lin et al., 2023) datasets for toxicity classification, and NELA-2018 dataset (Nørregaard et al., 2019) for misinformation detection. Toxicity classification is prone to bias, particularly as documents containing sexuality-related terms are often misclassified as toxic, resulting in an increased false positive rate. For the NELA-2018 dataset (Nørregaard et al., 2019), the sensitive attribute raising fairness concerns is the political leaning, either left or right, as indicated in (Park et al., 2022). All datasets are divided into 60%, 20%, and 20% splits for training, validation, and testing, respectively.

For preprocessing, we utilize tokenization and vectorization techniques to convert the raw text data into numerical representations suitable for machine learning models. We employ the SpaCy English tokenizer for tokenization, as discussed in (Honnibal and Montani, 2017), and the Doc2Vec model (Le and Mikolov, 2014) for vectorization, transforming the tokenized text into fixed-length feature vectors.

We conduct extensive experiments to validate the feasibility of our proposed Fairness-Aware Online PU learning as well as offline learning. Two different PU approaches, uPU and nnPU are implemented for three different classifiers, linear, MLP, and LSTM, where MLP consists of two hidden layers with 128 nodes in each layer in offline learning and 64 nodes in online learning. For LSTM, the hidden size is determined as 128. For both offline and online learning, we vary $\lambda_f \in \{10^{-2}, 10^{-1}, 10^{0}\}$ and report when the accuracy is the best. The surrogate function used for PU risk estimators is double hinge loss $\ell(z) = \max(-z, \max(0, \frac{1}{2} - \frac{1}{2}z))$, where $z = y \cdot f(x)$. In the offline setting, the training runs 50 epochs with an Adam optimizer and learning rate $\text{lr} = 0.001$. The batch size is 1024, and the hyperparameter in offline learning λ_r is 10^{-4}.

In the online setting, we conduct extensive experiments with the fixed total number of rounds $T = 200$. Naturally, the batch size in online learning is equal to the number of incoming samples at each round, i.e. $b = N/T$ where N is the total number of training samples. We vary the hyperparameter β by letting the initial step size $\eta_1 = b/(\beta \cdot \sqrt{1})$ be the level of learning rate $\eta_1 \in \{10^{-2}, 10^{-1}, 10^{0}\}$ for linear and MLP classifier, and $\eta_1 \in \{10^{2}, 10^{1}, 10^{0}\}$ for LSTM, while $\lambda_r = 0.01$ is fixed following (Zhang et al., 2021). In both offline and online learning, we run 10 experiments for each case to obtain the mean and standard deviation.

H Analysis in state-of-the-art PU methods (Robust-PU)

We also consider applying Robust-PU learning (Zhu et al., 2023), which is a state-of-the-art in PU learning literature.

Robust-PU generates weights for each sample by measuring 'hardness' recognizing easy positive samples and reliable negative samples. The positive-unlabeled samples are trained by weighted supervised learning,

$$R_{\text{robust}} = \mathbb{E}_p[\boldsymbol{w}_p^\top \cdot \ell(f(\boldsymbol{X}))] + \mathbb{E}_u[\boldsymbol{w}_n^\top \cdot \ell(-f(\boldsymbol{X}))] \tag{24}$$

where $\boldsymbol{w}_p$ and $\boldsymbol{w}_n$ denote weights for easy positive samples and reliable negative samples, respectively.

However, the assumption and mechanism in Robust-PU face significant challenges when applied to NLP datasets. Specifically, the ambiguity, context-dependence, and inherent noisiness of text data make it difficult to meet the requirements for reliable negative sample selection and accurate hardness measurement. These factors collectively hinder Robust-PU's performance in NLP, necessitating further adaptations and refinements to adress the unique challenges of textual data.

We validate the effectiveness in Robust-PU in tabular dataset, and ineffectiveness in NLP dataset.