A Appendix

A.1 Simplification of Equation 3

To facilitate the optimization of the objective function Eq. 3, we use D to denote $D(\mathbf{x}^{pu})$ and C to denote $C(\mathbf{x}^{pu})$ and simplify Eq. 3 to

$$\begin{aligned}&\min_C \max_D \ \mathrm{V}(D,C)\\&= \underbrace{\mathbb{E}_{\mathbf{x}^p\sim P^p(\mathbf{x}^p)}[\log D(\mathbf{x}^p)] + \mathbb{E}_{\mathbf{x}^u\sim P^u(\mathbf{x}^u)}[\log(1-D(\mathbf{x}^u))]}_{\text{IV}:-H(P^L,D(\mathbf{x}^{pu}))}\\&+\lambda\cdot\underbrace{\mathbb{E}_{\mathbf{x}^u\sim P^u(\mathbf{x}^u)}[\underbrace{(\log(1-C(\mathbf{x}^u))-\log(C(\mathbf{x}^u)))}_{\text{V}}(2D(\mathbf{x}^u)-1)]}_{\text{VI}}\end{aligned} \tag{9}$$

where P^p denotes the distribution of the positive data.

Simplification We now derive Eq. 9. Recall the loss function in Eq. 3:

$$\begin{aligned}&\min_C \max_D \ \mathrm{V}(D,C) = \underbrace{-\sum_{i=1}^{n} KL(P_i^{pu}||D_i^{pu})}_{\text{I}}\\&+\lambda\Big(\overbrace{\underbrace{\sum_{i=1}^{n_0} KL(D_i^u||C_i^u)}_{\text{II}} - \underbrace{\sum_{i=1}^{n_0} KL(D_i^u||\widetilde{C}_i^u)}_{\text{III}}}^{①}\Big)\end{aligned} \tag{10}$$

KL-divergence is defined as:

$$KL(P||Q) = \sum_{x\in\mathcal{X}} P(x)\log P(x) - P(x)\log Q(x) \tag{11}$$

$\mathcal{X}$ denotes the probability space and it is 1 or 0 ($\mathcal{X}=\{1,0\}$) in our case. We first address term I in Eq. 10. If we use D to denote $D_i^{pu}(1)$ (the probability for the ith instance being positive judged by discriminator) and P to denote $P_i^{pu}(1)$, then $D_i^{pu}(0) = 1-D$ and $P_i^{pu}(0) = 1-P$, then we have:

$$\begin{aligned}&-KL(P_i^{pu}||D_i^{pu})\\&= -P\log P + P\log D - (1-P)\log(1-P) + (1-P)\log(1-D)\\&= P\log D + (1-P)\log(1-D)\end{aligned} \tag{12}$$

Due to the fact that $P_i^{pu}(0)=0$ and $P_i^{pu}(1)=1$ if the ith instance is positive and $P_i^{pu}(1)=0$ and $P_i^{pu}(0)=1$ if the ith instance is unlabeled, we can rewrite the result as:

$$\mathbb{E}_{\mathbf{x}^p\sim P^p(\mathbf{x}^p)}[\log D(\mathbf{x}^p)] + \mathbb{E}_{\mathbf{x}^u\sim P^u(\mathbf{x}^u)}[\log(1-D(\mathbf{x}^u))] \tag{13}$$

Similarly, for term ① in Eq. 10, if we use D to denote $D_i^u(1)$ and C to denote $C_i^u(1)$, then we get:

$$\begin{aligned}&KL(D_i^u||C_i^u) - KL(D_i^u||(1-C_i^u))\\&= D\log D - D\log C + (1-D)\log(1-D)\\&\quad -(1-D)\log(1-C) - D\log D\\&\quad +D\log(1-C) - (1-D)\log(1-D) + (1-D)\log C\\&= -D\log C - (1-D)\log(1-C) + D\log(1-C) + (1-D)\log C\\&= (\log(1-C)-\log C)(2D-1)\end{aligned} \tag{14}$$

Then we have:

$$\begin{aligned}&(\log(1-C)-\log C)(2D-1)\\&=\mathbb{E}_{\mathbf{x}^u\sim P^u(\mathbf{x}^u)}[(\log(1-C(\mathbf{x}^u))-\log(C(\mathbf{x}^u)))(2D(\mathbf{x}^u)-1)]\end{aligned} \tag{15}$$

Combining Eqs. 13 and 15, we get Eq. 16:

$$\begin{aligned}&\min_C \max_D \ \mathrm{V}(D,C)\\&= \underbrace{\mathbb{E}_{\mathbf{x}^p\sim P^p(\mathbf{x}^p)}[\log D(\mathbf{x}^p)] + \mathbb{E}_{\mathbf{x}^u\sim P^u(\mathbf{x}^u)}[\log(1-D(\mathbf{x}^u))]}_{\text{IV}:-H(P^L,D(\mathbf{x}^{pu}))}\\&+\lambda\cdot\underbrace{\mathbb{E}_{\mathbf{x}^u\sim P^u(\mathbf{x}^u)}[\underbrace{(\log(1-C(\mathbf{x}^u))-\log(C(\mathbf{x}^u)))}_{\text{V}}(2D(\mathbf{x}^u)-1)]}_{\text{VI}}\end{aligned} \tag{16}$$

where P^p denotes the distribution of the positive data.

A.2 Theoretical Analysis About the Learned Classifier

In this section, we analyze the properties of the learned classifier and show why Eq. 3 can perform PU learning. Intuitively, from Eq. 10 (same as Eq. 3), we can see that $D(\cdot)$ is biased if we only consider term I because the unlabeled set contains both positive and negative examples. However, terms II and III help correct the bias. Eq. 16 (same as Eq. 7 in the paper), which is derived from Eq. 10 for training, shows the property more clearly than Eq. 10. Notice that the bias in term I of Eq. 10 will result in high precision and low recall for the positive class. Now back to Eq. 16 and let us imagine that most examples in the unlabeled set are regarded as negative by $C(\cdot)$ (meaning low recall). From Eq. 16, we can see that the value of term V will be below zero. But when optimizing $D(\cdot)$, term VI will push $D(\cdot)$ up for these data points, and thus the bias is reduced and the low recall problem is mitigated because in the next optimization iteration, $C(\cdot)$ will follow $D(\cdot)$ to go up for these data points.

We now theoretically discuss the optimal decision surface of the classifier $C(\cdot)$ learned by the proposed PAN.

Proposition 1. Let $T(\mathbf{x}) = \log[1-C(\mathbf{x})] - \log[C(\mathbf{x})]$, $\varepsilon(\mathbf{x}) = f(T(\mathbf{x}))$, the learned optimal decision surface of $C(\cdot)$ is:

$$\varepsilon(\mathbf{x}) = \frac{1}{2} - \frac{P^p(\mathbf{x})}{P^p(\mathbf{x})+P^u(\mathbf{x})} \tag{17}$$

$f(\cdot)$ is a type of function that satisfies $\varepsilon(\mathbf{x})\cdot T(x) > 0$.

From Proposition 1, we can see that PAN finds the decision surface by combining $P^p(\mathbf{x})$ and $P^u(\mathbf{x})$. The combination is controlled by $\varepsilon(\mathbf{x})$. $\varepsilon(\mathbf{x})$ is a function of $C(\mathbf{x})$. As the precise expression of the optimal decision surface is highly complex, here we only show its properties, i.e., Eq. 21.

Proof of Proposition 1 The training criterion for discriminator D, given any classifier C, is to maximize the quantity $V(C,D)$,

$$\begin{aligned}V(C,D) = &\int_{\mathbf{x}} P^p(\mathbf{x})\log D(\mathbf{x})dx + \int_{\mathbf{x}} P^u(\mathbf{x})\log(1-D(\mathbf{x}))dx\\&+\lambda\int_{\mathbf{x}} P^u(\mathbf{x})[\log(1-C(\mathbf{x})) - \log C(\mathbf{x})](2D(\mathbf{x})-1)dx\end{aligned} \tag{18}$$

Clearly, the maximum point appears at the point with derivative 0. Then we calculate the partial derivative of $V(C, D)$ to D and get:

$$\underbrace{\frac{P^p(\mathbf{x})}{D(\mathbf{x})} - \frac{P^u(\mathbf{x})}{1 - D(\mathbf{x})}}_{ⓐ} + \underbrace{\lambda P^u(\mathbf{x})[\log(1 - C(\mathbf{x})) - \log C(\mathbf{x})]}_{ⓑ} = 0 \tag{19}$$

Directly computing the solution is complex. If we omit term ⓑ for the time being, the solution for Eq. 19 is $D(\mathbf{x}) = \frac{P^p(\mathbf{x})}{P^p(\mathbf{x})+P^u(\mathbf{x})}$. After bringing back ⓑ, this solution should be revised as follows: With the definition of $T(\mathbf{x}) = \log[1 - C(\mathbf{x})] - \log[C(\mathbf{x})]$ and $\varepsilon(\mathbf{x}) = f(T(\mathbf{x}))$, we can re-write the solution after revision:

$$D^*(\mathbf{x}) = \frac{P^p(\mathbf{x})}{P^p(\mathbf{x}) + P^u(\mathbf{x})} + \varepsilon(\mathbf{x}) \tag{20}$$

where $\varepsilon(\mathbf{x})$ is a function of $T(\mathbf{x})$ since $P^p(\mathbf{x})$ and $P^u(\mathbf{x})$ are decided by the dataset and $\varepsilon(\mathbf{x})$ changes with the change of $T(\mathbf{x})$. The exact expression of $\varepsilon(\mathbf{x})$ is difficult but we can show $\varepsilon(\mathbf{x}) \propto T(\mathbf{x})$ in our case. In Eq. 19, term ⓐ decreases monotonously when $D(\mathbf{x}) \in (0, 1)$,[8] and both λ and $P^p(\mathbf{x})$ are greater than 0. In this case, if $T(\mathbf{x}) > 0$ ($T(\mathbf{x}) < 0$), to keep Eq. 19 equal to 0, $D(\mathbf{x})$ must move toward the positive (negative) direction, which indicates $\varepsilon(\mathbf{x}) > 0$ ($\varepsilon(\mathbf{x}) < 0$). Formally, we have:

$$\varepsilon(\mathbf{x}) \propto T(\mathbf{x}); \quad \varepsilon(\mathbf{x})T(\mathbf{x}) > 0 \tag{21}$$

Note that the training objective of D can be interpreted as using the training data ($P^p(\mathbf{x})$ and $P^u(\mathbf{x})$) to find a discrimination bound and utilizing the learned knowledge in C to adapt it. The mini-max game in Eq. 7 can now be reformulated as:

$$\begin{aligned} L(C) =& \max_D \ \mathrm{V}(C, D) \\ =& \mathbb{E}_{\mathbf{x}\sim P^u(\mathbf{x})}[(\log(1 - C(\mathbf{x})) - \log(C(\mathbf{x})))(2D^*(\mathbf{x}) - 1)] \\ =& \mathbb{E}_{\mathbf{x}\sim P^u(\mathbf{x}^u)}[T(\mathbf{x})(\frac{2P^p(\mathbf{x})}{P^p(\mathbf{x}) + P^u(\mathbf{x})} + 2\varepsilon(\mathbf{x}) - 1)] \end{aligned} \tag{22}$$

Clearly, because the range of $T(\mathbf{x})$ is $(-\epsilon, \epsilon)$, $L(C)$ achieves its minimum when $T(\mathbf{x})$ and $(2D^*(\mathbf{x}) - 1)$ have opposite signs. Then, the optimal $T^*(\mathbf{x})$ satisfies:

$$T^*(\mathbf{x})[\frac{2P^p(\mathbf{x})}{P^p(\mathbf{x}) + P^u(\mathbf{x})} + 2\varepsilon(\mathbf{x}) - 1] < 0 \tag{23}$$

As we introduced in Footnote 5, $C(\mathbf{x}) \in (0, 1)$. In this case, we use $C(\mathbf{x}) = 0.5$ as the decision surface to perform classification. Clearly, this decision surface equals to $T^*(\mathbf{x}) = \log(1 - 0.5) - \log(0.5) = 0$. In summary, we get the optimal decision surface:

$$\varepsilon(\mathbf{x}) = \frac{1}{2} - \frac{P^p(\mathbf{x})}{P^p(\mathbf{x}) + P^u(\mathbf{x})} \tag{24}$$

[8] We force $D(\mathbf{x})$ to satisfy the condition by adding a Sigmoid function to the end of D. We also force the output range of C into $(0, 1)$ using the same method.

A.3 Plots of Precision, Recall, and Accuracy

Figures 2-7 show the test accuracy, precision and recall in each epoch of each method. The usual first-order exponential weighted moving average smoothing with weight 0.7 is applied to the figures. To save space, we only show the curves of 100 epochs, but the final results in Table 1 in the paper are produced with more epochs as NNPU takes slightly longer to reach the peak (see below). Based on the Figures, we can make the following observation.

(1). PAN and NNPUSB are robust across all datasets with both high precision and high recall. And clearly PAN is also better than NNPUSB. NNPU is rather unbalanced for precision and recall for YELP and RT, either very high precision but very low recall, or vice versa. a-GAN and UPU have the same problem, which is highly undesirable. PAN also outperforms baselines consistently in accuracy for all six datasets.

(2). a-GAN is unstable. Precision, recall, and accuracy fluctuate greatly from one epoch to the next. Stability problem of GAN is well known. Our adaptation requiring reinforcement learning to train it is likely to have made the problem worse.

(3). NNPU converges slowly (Figure 3). It didn't converge even at 100 epochs. We report its best accuracy and F-score in Table 1 in 200 epochs (it converged earlier than 200).

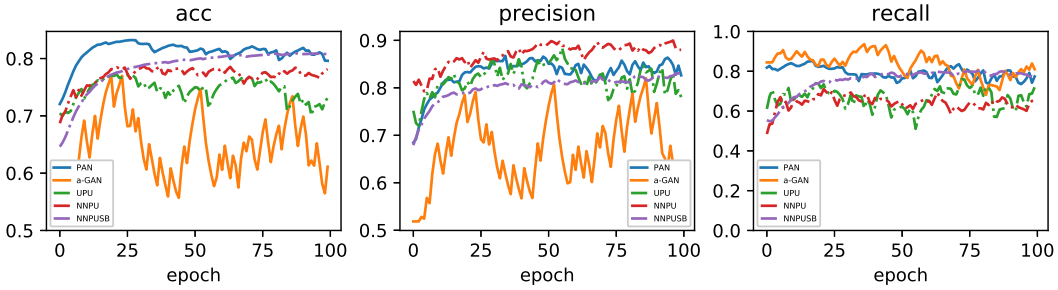

Figure 2: YELP - to save space, we only show 100 epochs in this and the figures below.

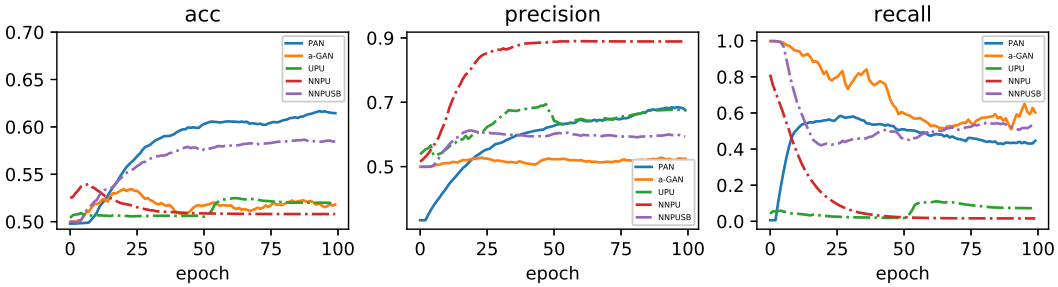

Figure 3: RT

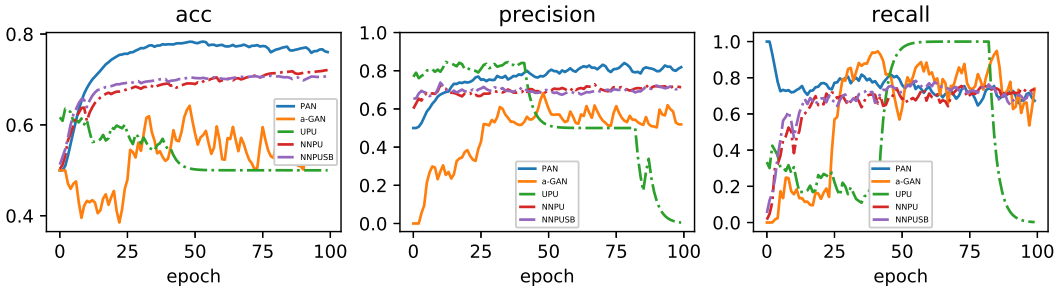

Figure 4: IMDB

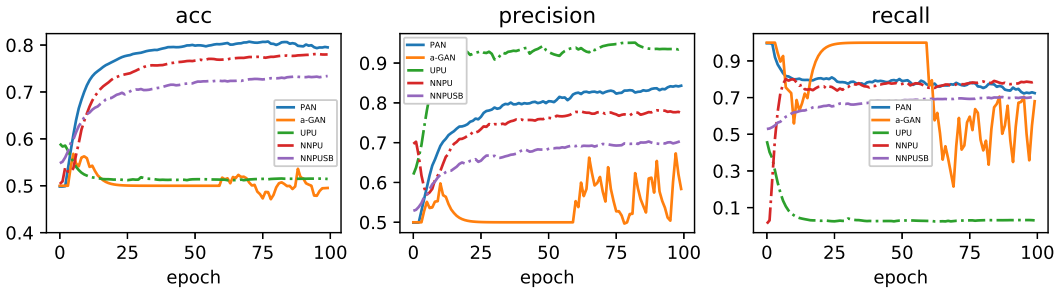

Figure 5: 20News

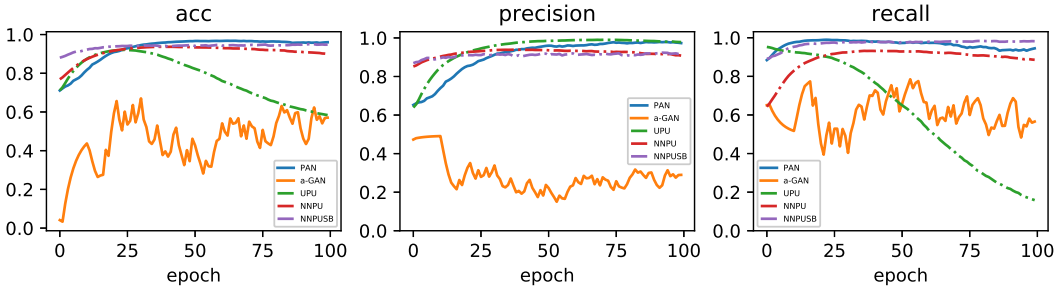

Figure 6: MNIST

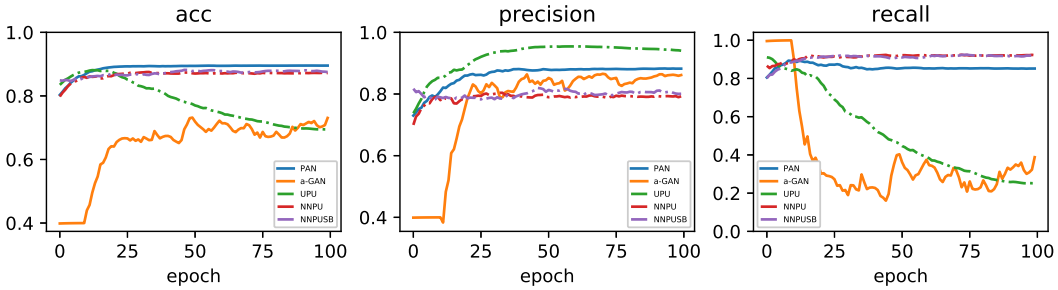

Figure 7: CIFAR10