

When AUC meets DRO: Optimizing Partial AUC for Deep Learning with Non-Convex Convergence Guarantee

Dixian Zhu, Gang Li, Bokun Wang, Xiaodong Wu,
Tianbao Yang

Motivation

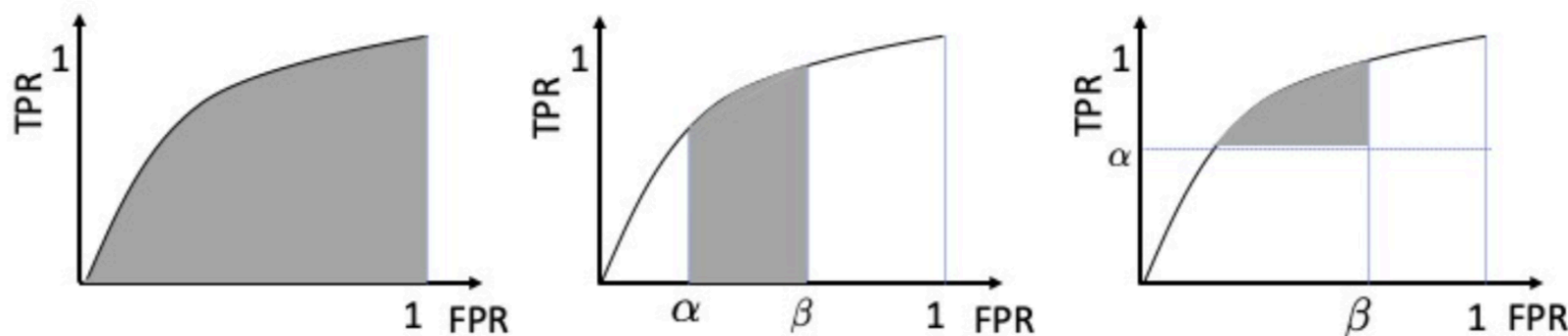


Figure 1: From left to right: AUC, one-way pAUC, two-way pAUC

There are few rigorous and efficient algorithms developed for pAUC maximization for deep learning. Our pAUC maximization method with distributional robust optimization (DRO) technique can provide the **exact&soft estimators** and **convergence guarantee**.

Preliminaries

- Non-parametric estimator

- One way Partial AUC (OPAUC)

$$\widehat{\text{OPAUC}}(h, \alpha_0, \alpha_1) = \frac{1}{n_+} \frac{1}{n_-} \sum_{\mathbf{x}_i \in \mathcal{S}_+} \sum_{\mathbf{x}_j \in \mathcal{S}_-^\downarrow[k_1+1, k_2]} \mathbb{I}(h(\mathbf{x}_i) > h(\mathbf{x}_j))$$

where $k_1 = \lceil n_- \alpha_0 \rceil$, $k_2 = \lfloor n_- \alpha_1 \rfloor$. In this work, we will focus on optimizing $\widehat{\text{OPAUC}}(h, 0, \beta)$ for some $\beta \in (0, 1)$.

- Two way Partial AUC (TPAUC)

$$\widehat{\text{TPAUC}}(h, \alpha, \beta) = \frac{1}{n_+} \frac{1}{n_-} \sum_{\mathbf{x}_i \in \mathcal{S}_+^\uparrow[1, k_1]} \sum_{\mathbf{x}_j \in \mathcal{S}_-^\downarrow[1, k_2]} \mathbb{I}(h(\mathbf{x}_i) > h(\mathbf{x}_j))$$

where $k_1 = \lfloor n_+ \alpha \rfloor$, $k_2 = \lfloor n_- \beta \rfloor$.

Preliminaries

Distributionally Robust Optimization (DRO). For a set of random loss functions $\ell_1(\cdot), \dots, \ell_n(\cdot)$, a DRO loss can be written as

$$\hat{L}_\phi(\cdot) = \max_{\mathbf{p} \in \Delta} \sum_j p_j \ell_j(\cdot) - \lambda D_\phi(\mathbf{p}, 1/n), \quad (4)$$

Lemma 1. *By using KL divergence measure, we have*

$$\hat{L}_{kl}(\cdot; \lambda) = \lambda \log \left(\frac{1}{n} \sum_{i=1}^n \exp \left(\frac{\ell_i(\cdot)}{\lambda} \right) \right). \quad (5)$$

By using the CVaR divergence $\phi_c(t)$ for some γ such that $n\gamma$ is an integer, we have,

$$\hat{L}_{cvar}(\cdot; \gamma) = \frac{1}{n\gamma} \sum_{i=1}^{n\gamma} \ell_{[i]}(\cdot), \quad (6)$$

where $\ell_{[i]}(\cdot)$ denotes the i -th largest value in $\{\ell_1, \dots, \ell_n\}$.

Methods for OPAUC

- Let $L(\mathbf{w}, \mathbf{x}_i, \mathbf{x}_j) = \ell(h_{\mathbf{w}}(\mathbf{x}_i) - h_{\mathbf{w}}(\mathbf{x}_j))$, the surrogate loss:

$$\min_{\mathbf{w}} \frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{S}_+} \frac{1}{n_- \beta} \sum_{\mathbf{x}_j \in \mathcal{S}_-^\downarrow[1, n_- \beta]} L(\mathbf{w}; \mathbf{x}_i, \mathbf{x}_j), \quad (7)$$

- CVaR-based (exact) estimator:

$$\min_{\mathbf{w}} \frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{S}_+} \hat{L}_\phi(\mathbf{w}; \mathbf{x}_i). \quad (8)$$

Theorem 1. By choosing $\phi(\cdot) = \phi_c(\cdot) = \mathbb{I}(\cdot \in (0, 1/\beta])$, then the problem (8) is equivalent to

$$\min_{\mathbf{w}} \min_{\mathbf{s} \in \mathbb{R}^{n_+}} F(\mathbf{w}, \mathbf{s}) = \frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{S}_+} \left(s_i + \frac{1}{\beta} \psi_i(\mathbf{w}, s_i) \right), \quad (9)$$

where $\psi_i(\mathbf{w}, s_i) = \frac{1}{n_-} \sum_{\mathbf{x}_j \in \mathcal{S}_-} (L(\mathbf{w}; \mathbf{x}_i, \mathbf{x}_j) - s_i)_+$. If ℓ is a monotonically decreasing function for $\ell(\cdot) > 0$, then the objective in (8) is equivalent to (7) of OPAUC.

Methods for OPAUC

Algorithm 1 SOPA

- 1: Set $s^1 = 0$ and initialize $\mathbf{w}$
- 2: **for** $t = 1, \dots, T$ **do**
- 3: Sample two mini-batches $\mathcal{B}_+ \subset \mathcal{S}_+, \mathcal{B}_- \subset \mathcal{S}_-$
- 4: Let $p_{ij} = \mathbb{I}(\ell(h(\mathbf{w}_t, \mathbf{x}_i) - h(\mathbf{w}_t, \mathbf{x}_j)) - s_i^t > 0)$
- 5: Update $s_i^{t+1} = s_i^t - \frac{\eta_2}{n_+} (1 - \frac{\sum_j p_{ij}}{\beta |\mathcal{B}_-|})$ for $\mathbf{x}_i \in \mathcal{B}_+$
- 6: Compute a gradient estimator ∇_t by

$$\nabla_t = \frac{1}{\beta |\mathcal{B}_+| |\mathcal{B}_-|} \sum_{\mathbf{x}_i \in \mathcal{B}_+} \sum_{\mathbf{x}_j \in \mathcal{B}_-} p_{ij} \nabla_{\mathbf{w}} L(\mathbf{w}_t; \mathbf{x}_i, \mathbf{x}_j)$$

- 7: Update $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_1 \nabla_t$
 - 8: **end for**
-

Methods for OPAUC

- KLDRO-based (soft) estimator:

$$\min_{\mathbf{w}} \frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{S}_+} \hat{L}_\phi(\mathbf{w}; \mathbf{x}_i). \quad (8)$$

Theorem 2. By choosing $\phi(\cdot) = \phi_{kl}(\cdot)$, then the problem (8) becomes

$$\min_{\mathbf{w}} \frac{1}{n_+} \sum_{\mathbf{x}_i \sim \mathcal{S}_+} \lambda \log \mathbb{E}_{\mathbf{x}_j \in \mathcal{S}_-} \exp\left(\frac{L(\mathbf{w}; \mathbf{x}_i, \mathbf{x}_j)}{\lambda}\right). \quad (10)$$

If $\ell(\cdot)$ is a monotonically decreasing function for $\ell(\cdot) > 0$, when $\lambda = 0$, the above objective is a surrogate of $\widehat{\text{OPAUC}}(h_{\mathbf{w}}, 0, \frac{1}{n_-})$; and when $\lambda = +\infty$, the above objective is a surrogate of $\widehat{\text{OPAUC}}(h_{\mathbf{w}}, 0, 1)$, i.e., the AUC.

Methods for OPAUC

Algorithm 2 SOPA-s

- 1: Set $\mathbf{u}^1 = 0$ and initialize $\mathbf{w}$
- 2: **for** $t = 1, \dots, T$ **do**
- 3: Sample two mini-batches $\mathcal{B}_+ \subset \mathcal{S}_+, \mathcal{B}_- \subset \mathcal{S}_-$
- 4: For each $\mathbf{x}_i \in \mathcal{B}_+$, update $u_i^{t+1} = (1 - \gamma_0)u_i^t + \gamma_0 \frac{1}{|\mathcal{B}_-|} \sum_{\mathbf{x}_j \in \mathcal{B}_-} \exp\left(\frac{L(\mathbf{w}_t; \mathbf{x}_i, \mathbf{x}_j)}{\lambda}\right)$
- 5: Let $p_{ij} = \exp(L(\mathbf{w}_t; \mathbf{x}_i, \mathbf{x}_j)/\lambda)/u_i^t$
- 6: Compute a gradient estimator ∇_t by

$$\nabla_t = \frac{1}{|\mathcal{B}_+|} \frac{1}{|\mathcal{B}_-|} \sum_{\mathbf{x}_i \in \mathcal{B}_+} \sum_{\mathbf{x}_j \in \mathcal{B}_-} p_{ij} \nabla L(\mathbf{w}_t; \mathbf{x}_i, \mathbf{x}_j)$$

- 7: Update $\mathbf{v}_t = (1 - \gamma_1)\mathbf{v}_{t-1} + \gamma_1 \nabla_t$
 - 8: Update $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{v}_t$ (or Adam-style)
 - 9: **end for**
-

Methods for OPAUC

Theorem 3. *Under Assumption 2, Algorithm 1 ensures that after $T = O(1/(\beta\epsilon^4))$ iterations we can find an ϵ nearly stationary solution of $F(\mathbf{w}, \mathbf{s})$, i.e., $\mathbb{E}\|\nabla F_{\hat{\rho}}(\mathbf{w}_{\tau}, \mathbf{s}_{\tau})\|^2 \leq \epsilon^2$ for a randomly selected $\tau \in \{1, \dots, T\}$ and $\hat{\rho} = 1.5\rho$.*

Theorem 4. *Under Assumption 2, Algorithm 2 with $\gamma_0 = O(B_- \epsilon^2)$, $\gamma_1 = O(\min\{B_-, B_+\} \epsilon^2)$, $\eta = O(\min\{\gamma_0 B_1/n_+, \gamma_1\})$ ensures that after $T = O(\frac{1}{\min(B_+, B_-) \epsilon^4} + \frac{n_+}{B_+ B_- \epsilon^4})$ iterations we can find an ϵ -stationary solution of $F(\mathbf{w})$, i.e., $\mathbb{E}[\|\nabla F(\mathbf{w}_{\tau})\|^2] \leq \epsilon^2$ for a randomly selected $\tau \in \{1, \dots, T\}$, where $B_+ = |\mathcal{B}_+|$ and $B_- = |\mathcal{B}_-|$.*

Methods for TPAUC

- KLDRO-based (soft) estimator:

$$F(\mathbf{w}; \phi, \phi') = \max_{\mathbf{p} \in \Delta} \sum_{\mathbf{x}_i \in \mathcal{S}_+} p_i \hat{L}_\phi(\mathbf{x}_i, \mathbf{w}) - \lambda' D_{\phi'}(\mathbf{p}, \frac{1}{n_+}).$$

Lemma 3. *When $\phi = \phi' = \phi_{kl}$, we have*

$$\begin{aligned} F(\mathbf{w}; \phi_{kl}, \phi_{kl}) \\ = \lambda' \log E_{\mathbf{x}_i \sim \mathcal{S}_+} \left(E_{\mathbf{x}_j \sim \mathcal{S}_-} \exp\left(\frac{\ell(\mathbf{w}; \mathbf{x}_i, \mathbf{x}_j)}{\lambda}\right) \right)^{\frac{\lambda}{\lambda'}}. \end{aligned}$$

For minimizing this function, we formulate the problem as a novel three-level compositional stochastic optimization:

$$\min_{\mathbf{w}} f_1\left(\frac{1}{n_+} \sum_{\mathbf{x}_i \in \mathcal{S}_+} f_2(g_i(\mathbf{w}))\right),$$

Methods for TPAUC

Algorithm 3 SOTA-s

- 1: Set $\mathbf{u}_0 = 0, v_0 = 0, \mathbf{m}_0 = 0$ and initialize $\mathbf{w}$
- 2: **for** $t = 1, \dots, T$ **do**
- 3: Sample two mini-batches $\mathcal{B}_+ \subset \mathcal{S}_+, \mathcal{B}_- \subset \mathcal{S}_-$
- 4: For each $\mathbf{x}_i \in \mathcal{B}_+$ compute $u_t^i = (1 - \gamma_0)u_{t-1}^i + \gamma_0 \frac{1}{|\mathcal{B}_-|} \sum_{\mathbf{x}_j \in \mathcal{B}_-} L(\mathbf{w}_t; \mathbf{x}_i, \mathbf{x}_j)$
- 5: Let $v_t = (1 - \gamma_1)v_{t-1} + \gamma_1 \frac{1}{|\mathcal{B}_+|} \sum_{\mathbf{x}_i \in \mathcal{B}_+} f_2(u_{t-1}^i)$
- 6: Let $p_{ij} = (u_{t-1}^i)^{\lambda/\lambda'-1} \exp(L(\mathbf{w}_t, \mathbf{x}_i, \mathbf{x}_j)/\lambda)/v_t$
- 7: Compute a gradient estimator ∇_t by

$$\nabla_t = \frac{1}{|\mathcal{B}_+|} \frac{1}{|\mathcal{B}_-|} \sum_{\mathbf{x}_i \in \mathcal{B}_+} \sum_{\mathbf{x}_j \in \mathcal{B}_-} p_{ij} \nabla L(\mathbf{w}_t; \mathbf{x}_i, \mathbf{x}_j)$$

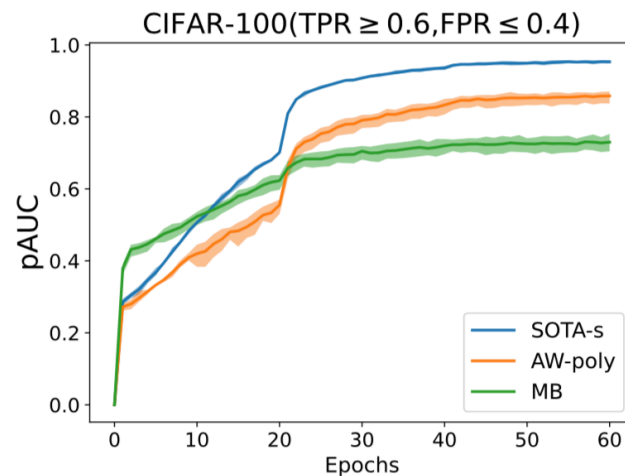
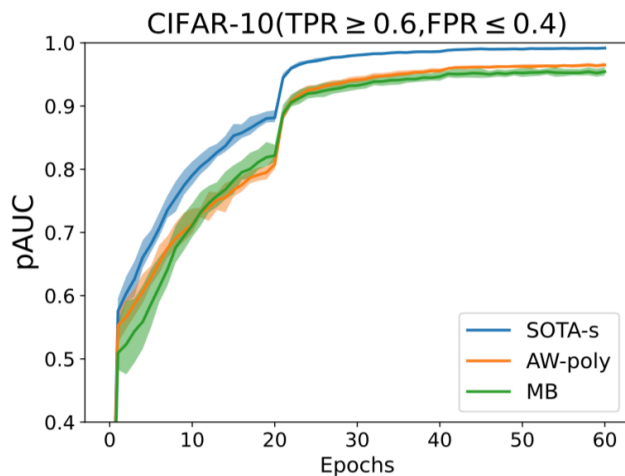
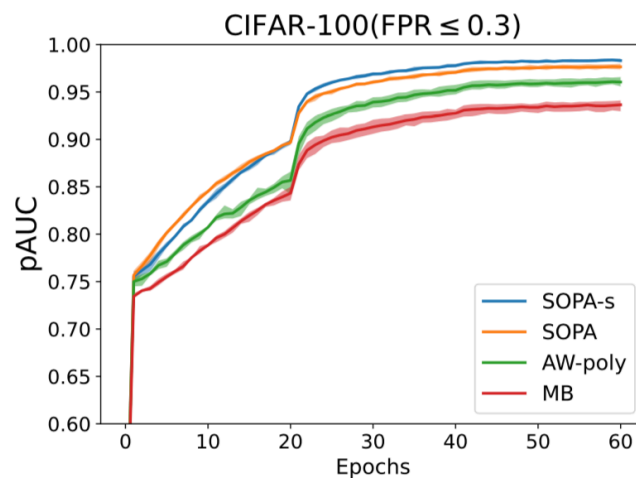
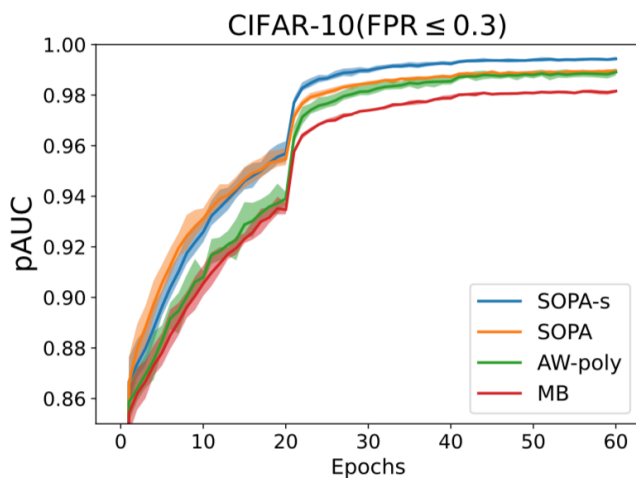
- 8: Update $\mathbf{m}_t = (1 - \gamma_2)\mathbf{m}_{t-1} + \gamma_2 \nabla_t$
 - 9: Update $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \mathbf{m}_t$ (or Adam-style)
 - 10: **end for**
-

Methods for TPAUC

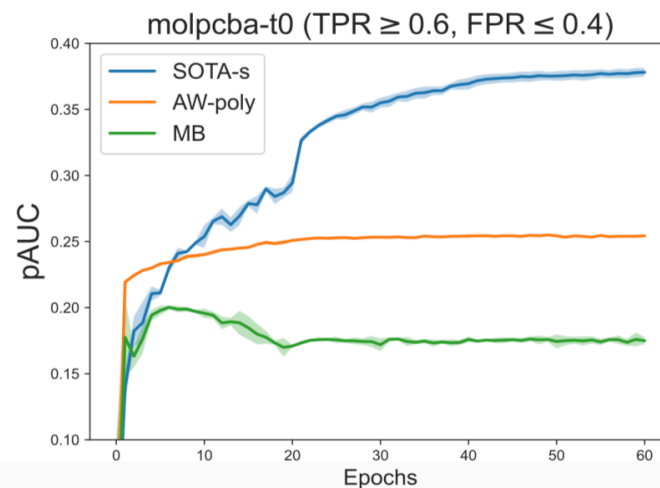
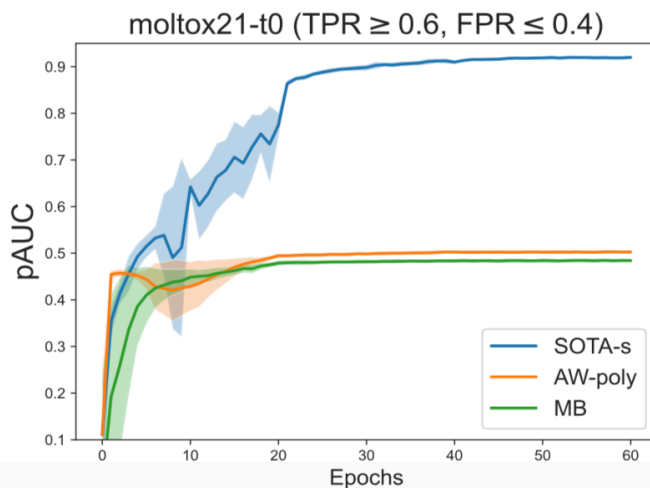
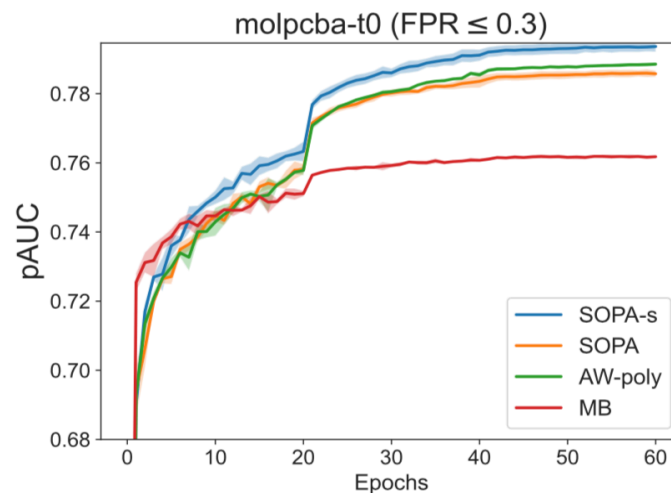
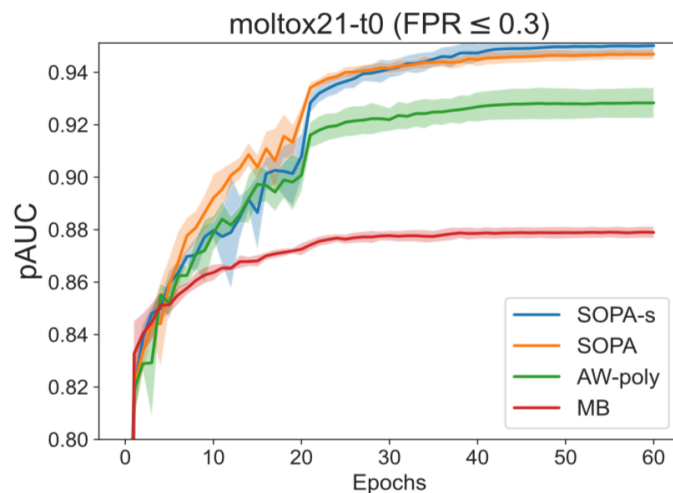
Theorem 5. *Under Assumption 2, Algorithm 3 with $\gamma_0 = O(B_- \epsilon^2)$, $\gamma_1 = O(B_+ \epsilon^2)$, $\gamma_2 = O(\min\{B_-, B_+\} \epsilon^2)$, $\eta = O(\min\{\gamma_0 B_1/n_+, \gamma_1, \gamma_2\})$ ensures that after $T = O(\frac{1}{\min(B_+, B_-) \epsilon^4} + \frac{n_+}{B_+ B_- \epsilon^4})$ iterations we can find an ϵ nearly stationary solution of $F(\mathbf{w})$, where $B_+ = |\mathcal{B}_+|$ and $B_- = |\mathcal{B}_-|$.*

Remark: It is notable that SOTA-s has an iteration complexity in the same order of SOPA-s for OPAUC maximization.

Experiments: training



Experiments: training



Experiments: testing

Table 1. One way partial AUC on testing data of three image datasets

Methods	CIFAR-10		CIFAR-100		Melanoma	
	FPR $\leq$ 0.3	FPR $\leq$ 0.5	FPR $\leq$ 0.3	FPR $\leq$ 0.5	FPR $\leq$ 0.3	FPR $\leq$ 0.5
CE	0.8446(0.0018)	0.8777(0.0014)	0.7338(0.0047)	0.7787(0.0044)	0.7651(0.0135)	0.8151(0.0028)
AUC-SH	0.8657(0.0056)	0.8948(0.0036)	0.7467(0.0047)	0.7930(0.0027)	0.7824(0.0138)	0.8176(0.0160)
AUC-M	0.8678(0.0016)	0.8934(0.0022)	0.7371(0.0031)	0.7828(0.0005)	0.7788(0.0068)	0.8249(0.0141)
P-push	0.8610(0.0007)	0.8889(0.0021)	0.7445(0.0025)	0.7930(0.0029)	0.7440(0.0130)	0.8028(0.0170)
MB	0.8690(0.0016)	0.8931(0.0015)	0.7487(0.0017)	0.7930(0.0014)	0.7683(0.0303)	0.8184(0.0278)
AW-poly	0.8664(0.0052)	0.8915(0.0075)	0.7490(0.0058)	0.7909(0.0068)	0.7936(0.0238)	0.8355(0.0067)
SOPA	0.8766(0.0034)	0.9028(0.0031)	0.7551(0.0044)	0.7999(0.0028)	0.8093(0.0248)	0.8585(0.0210)
SOPA-s	0.8691(0.0036)	0.8961(0.0036)	0.7468(0.0056)	0.7877(0.0053)	0.7775(0.0076)	0.8401(0.0206)

Table 2. Two way partial AUC on testing data of three image datasets; (α, β) represents $\text{TPR} \geq \alpha$ and $\text{FPR} \leq \beta$.

Methods	CIFAR-10		CIFAR-100		Melanoma	
	(0.6,0.4)	(0.5,0.5)	(0.6,0.4)	(0.5,0.5)	(0.6,0.4)	(0.5,0.5)
CE	0.4981(0.0078)	0.6414(0.0080)	0.2178(0.0136)	0.4011(0.0118)	0.3399(0.0135)	0.5150(0.0038)
AUC-SH	0.5622(0.0064)	0.6923(0.0071)	0.2599(0.0061)	0.4397(0.0062)	0.3640(0.0354)	0.5291(0.0312)
AUC-M	0.5691(0.0021)	0.6907(0.0125)	0.2336(0.0041)	0.4153(0.0022)	0.3665(0.0646)	0.5404(0.0545)
P-push	0.5477(0.0077)	0.6781(0.0055)	0.2623(0.0042)	0.4417(0.0092)	0.3317(0.0304)	0.4870(0.0443)
MB	0.5404(0.0041)	0.6724(0.0011)	0.2207(0.0033)	0.4017(0.0149)	0.3330(0.0258)	0.4981(0.0252)
AW-poly	0.5536(0.0196)	0.6814(0.0203)	0.2489(0.0166)	0.4342(0.0112)	0.3878(0.0292)	0.5216(0.0288)
SOTA-s	0.5799(0.0202)	0.7074(0.0145)	0.2708(0.0055)	0.4528(0.0069)	0.4198(0.0825)	0.5865(0.0664)

Experiments: testing

Table 3. One way partial AUC on testing data of three molecular datasets

Methods	moltox21(t0)		molmuv(t1)		molpcba(t0)	
	FPR $\leq$ 0.3	FPR $\leq$ 0.5	FPR $\leq$ 0.3	FPR $\leq$ 0.5	FPR $\leq$ 0.3	FPR $\leq$ 0.5
CE	0.6671(0.0009)	0.6954(0.005)	0.8008(0.0090)	0.8201(0.0061)	0.6802(0.0002)	0.7169(0.0002)
AUC-SH	0.7161(0.0043)	0.7295(0.0036)	0.7880(0.0382)	0.8025(0.0437)	0.6939(0.0009)	0.7350(0.0015)
AUC-M	0.6866(0.0048)	0.7080(0.0020)	0.7960(0.0123)	0.8076(0.0175)	0.6985(0.0016)	0.7399(0.0005)
P-push	0.6946(0.0107)	0.7160(0.0073)	0.7832(0.0220)	0.7940(0.0321)	0.6841(0.0007)	0.7293(0.0043)
MB	0.7398(0.0131)	0.7329(0.0099)	0.7672(0.0563)	0.7772(0.0547)	0.6899(0.0002)	0.7253(0.0006)
AW-poly	0.7227(0.0024)	0.7271(0.0112)	0.7754(0.0372)	0.7883(0.0431)	0.6975(0.0006)	0.7350(0.0015)
SOPA	0.7209(0.0063)	0.7318(0.0084)	0.8187(0.0319)	0.8245(0.0312)	0.6989(0.0022)	0.7371(0.0011)
SOPA-s	0.7309(0.0151)	0.7330(0.0073)	0.8449(0.0399)	0.8412(0.0447)	0.7027(0.0018)	0.7416(0.0006)

Table 4. Two way partial AUC on testing data of three molecular datasets; (α, β) represents $\text{TPR} \geq \alpha$ and $\text{FPR} \leq \beta$.

Methods	moltox21(t0)		molmuv(t1)		molpcba(t0)	
	(0.6,0.4)	(0.5,0.5)	(0.6,0.4)	(0.5,0.5)	(0.6,0.4)	(0.5,0.5)
CE	0.0674(0.0014)	0.2082(0.0011)	0.1613(0.0337)	0.4691(0.0183)	0.0949(0.0006)	0.2639(0.0006)
AUC-SH	0.0640(0.0080)	0.2170(0.0140)	0.2600(0.1300)	0.4440(0.1280)	0.1400(0.0030)	0.3120(0.0030)
AUC-M	0.0660(0.0090)	0.2090(0.0100)	0.1140(0.0790)	0.4330(0.0530)	0.1420(0.0090)	0.3130(0.0030)
P-push	0.0610(0.0180)	0.2070(0.0120)	0.1860(0.1520)	0.4170(0.1080)	0.1350(0.0020)	0.3000(0.0120)
MB	0.0670(0.0150)	0.2150(0.0230)	0.1730(0.1530)	0.4260(0.1180)	0.0950(0.0020)	0.2620(0.0030)
AW-poly	0.0640(0.0100)	0.2060(0.0250)	0.1720(0.1440)	0.3930(0.1230)	0.1100(0.0010)	0.2810(0.0020)
SOTA-s	0.0680(0.0180)	0.2300(0.0210)	0.3270(0.1640)	0.5260(0.1220)	0.1430(0.0010)	0.3140(0.0020)

Thank you!