# Appendix of Domain Specified Optimization

## 1. Algorithm of DSO

Recalling the empirical total objective function of DSO in the following formulation:

$$\mathcal{L}_{Total} = \sum_{i=1}^{n_b} \ell(f(\mathbf{X}_i^S \mid \theta), \mathbf{Y}_i^S) + K(D_{\phi_k}(P_0 \parallel \mathbb{1}/n_b)) \sup_{U \in \mathcal{U}_{\mathbb{1}/n_b}} \sum_{i=1}^{n_b} U_i \ell(f(\mathbf{X}_i^S \mid \theta), \mathbf{Y}_i^{err}), \tag{1}$$

Then we present detailed training procedure of DSO in Alg. 1. We note that the formulation of DRO methods in [1, 2] implicitly models the shifted marginal distribution $P_U(\mathbf{X})$ via projecting the loss vector onto the divergence ball. Detailed derivation of such technique is present in [7]. Subsequently, we calculate both the DSO and the supervised loss on the training data $P_S(\mathbf{X})$. To further stabilize the training process, we sample the training data for twice (two batches), while the supervised loss is calculated on the former batch and the DSO loss is calculated on the latter one. To calculate the DSO loss, we first compute the projection of the loss vector on the divergence ball as $\{U_i\}_{i=1}^{n_b}$. We then construct the error label by a translation operation on $\mathbf{Y}^{S2}$. Meanwhile, we calculate the scaling term $\alpha = K(D_2(\mathbf{U} \| \mathbb{1}/n_b))$ by computing the $\chi^2$ divergence between $\mathbf{U}$ and $\mathbb{1}/n_b$ (supported in many python interfaces, e.g., scipy.stats.power_divergence), where $K(t) = 1.0$ when $t \leq 0.2$ and $K(t) = t$ otherwise. We fix the function $K(t)$ throughout our experiments. Finally, the total loss is computed and the network $f$ is updated.

**Algorithm 1** Training process of DSO

**Input**: The training distribution $P_S(\mathbf{X}, \mathbf{Y})$, the radius parameter $\rho$, the deep network $f$ parameterized by $\theta$, the initial learning rate as $\gamma$, the batch size number $n_b$, the total training number of iterations as $T$.

**for** $t = 1, 2, \ldots, T$ **do**
**1: Sample** $\{\mathbf{X}_i^{S1}, \mathbf{Y}_i^{S1}\}_{i=1}^{n_b}$ and $\{\mathbf{X}_i^{S2}, \mathbf{Y}_i^{S2}\}_{i=1}^{n_b}$ from $P_S$;
**2: Compute** the supervised loss $\mathcal{L}_{sup}$ by $\mathcal{L}_{sup} = \frac{1}{n_b} \sum_{i=1}^{n_b} \ell(f(\mathbf{X}_i^{S1} \mid \theta), \mathbf{Y}_i^{S1})$;
**3: Calculate** the loss vector of $\{\mathbf{X}_i^{S2}, \mathbf{Y}_i^{S2}\}_{i=1}^{n_b}$ as $\{\ell(f(\mathbf{X}_i^{S2} \mid \theta), \mathbf{Y}_i^{S2})\}_{i=1}^{n_b}$;
**4: Compute** the projection of $\{\ell(f(\mathbf{X}_i^{S2} \mid \theta), \mathbf{Y}_i^{S2})\}_{i=1}^{n_b}$ on the ball $\mathcal{U}_{\mathbb{1}/n_b}$ as $\{U_i\}_{i=1}^{n_b}$;
**5: Construct** the error label $\mathbf{Y}_i^{err} = \mathbf{Y}_i^{S2} + 1$;
**6: Calculate** the scaling term $\alpha$ via $\alpha = K(D_2(\mathbf{U} \| \mathbb{1}/n_b))$;
**7: Compute** the DSO loss $\mathcal{L}_{dso}$ by $\mathcal{L}_{dso} = \frac{\alpha}{n_b} \sum_{i=1}^{n_b} U_i \ell(f(\mathbf{X}_i^{S2} \mid \theta), \mathbf{Y}_i^{err})$;
**8: Compute** the total loss as $\mathcal{L}_{total} = \mathcal{L}_{sup} + \mathcal{L}_{dso}$;
**9: Update** the network parameter by $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}_{total}$;

**Output**: The trained deep network $f(\theta)$.

For TDSO, since we have unauthorized features, the uncertainty set is reduced into a point $P_U(\mathbf{X})$. It is noteworthy that the construction of $\mathbf{Y}^{err}$ in TDSO slightly differs from that in DSO. More specifically, TDSO first computes the pseudo-labels $\overline{\mathbf{Y}}_i^U$ on $\{\mathbf{X}_i^U\}_{i=1}^{n_b}$ from $f$ in Step 4, where $j \in \{1, 2, \ldots, C\}$ is the class index. TDSO then performs the translation on $\overline{\mathbf{Y}}_i^U$ to obtain $\mathbf{Y}^{err}$ such that the confidence of $f$ on $P_U$ reduces.

**Algorithm 2** Training process of TDSO

**Input**: The training distribution $P_S(\mathbf{X}, \mathbf{Y})$, the unauthorized distribution $P_U(\mathbf{X})$ the radius parameter $\rho$, the deep network $f$ parameterized by $\theta$, the initial learning rate as $\gamma$, the batch size number $n_b$, the total training number of iterations as $T$.

**for** $t = 1, 2, \ldots, T$ **do**
**1: Sample** $\{\mathbf{X}_i^S, \mathbf{Y}_i^S\}_{i=1}^{n_b}$ from $P_S$;
**2: Sample** $\{\mathbf{X}_i^U\}_{i=1}^{n_b}$ from $P_U$;
**3: Compute** the supervised loss $\mathcal{L}_{sup}$ by $\mathcal{L}_{sup} = \frac{1}{n_b}\sum_{i=1}^{n_b} \ell(f(\mathbf{X}_i^S \mid \theta), \mathbf{Y}_i^S)$;
**4: Compute** the pseudo labels of $f$ on $\{\mathbf{X}_i^U\}_{i=1}^{n_b}$ as $\overline{\mathbf{Y}}_i^U = \arg\max_j f(\mathbf{X}_i^U \mid \theta)$;
**5: Construct** the error label $\mathbf{Y}_i^{err} = \overline{\mathbf{Y}}_i^U + 1$;
**7: Compute** the TDSO loss $\mathcal{L}_{tdso}$ by $\mathcal{L}_{tdso} = \frac{1}{n_b}\sum_{i=1}^{n_b} \ell(f(\mathbf{X}_i^U \mid \theta), \mathbf{Y}_i^{err})$;
**8: Compute** the total loss as $\mathcal{L}_{total} = \mathcal{L}_{sup} + \mathcal{L}_{tdso}$;
**9: Update** the network parameter by $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}_{total}$;
**Output**: The trained deep network $f(\theta)$.

## 1.1. Theoretical proof

## 1.2. Convergence of DSO

Following the technique in [2, 7], we derive the concentration result of our method in its dual form. More specifically, we split the total loss function into the DSO loss and the supervised loss as follows:

$$\begin{cases} \mathcal{L}_{SUP} = \mathbb{E}_{P_S}[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})] \\ \mathcal{L}_{DSO} = K(D_{\phi_k}(P_0 \parallel P_S)) \sup_{P_U \in \mathcal{U}_{P_S}} \mathbb{E}_{P_U}[\ell(f(\mathbf{X} \mid \theta), \mathbf{Y}^{err})], \end{cases} \tag{2}$$

We first derive the concentration result of $\mathcal{L}_{DSO}$. For convenience, we denote the expected and empirical risks of $\mathcal{L}_{DSO}$ in (2) as $\mathcal{R}_k(\ell(\theta); P^S)$ and $\widehat{\mathcal{R}}_k(\ell(\theta); \widehat{P}^S)$, respectively. We start from the dual form of $\mathcal{R}_k(\ell(\theta); P^S)$ by the following lemma:

**Lemma 1.1.** *For any probability $P$ with $k_* = k/(k-1)$, $\rho \geq 0$, $p_k = (1 + k(k-1)\rho)^{\frac{1}{k}}$ and $s_k = K(D_f(P_0 \parallel P_S))$, the following dual form of $\mathcal{L}_{DSO}$ in* (2) *holds with strong duality:*

$$\mathcal{R}_k(\ell(f(\mathbf{X}^S \mid \theta); P) = s_k \inf_{\eta \in \mathbb{R}} \left\{ p_k \mathbb{E}_P \left[ (\ell(f(\mathbf{X}^S \mid \theta), \mathbf{Y}^{err}) - \eta)_+^{k_*} \right]^{\frac{1}{k_*}} + \eta \right\} \tag{3}$$

*, where the $\eta$ controls the "hardness" of the sample.*

The above lemma is immediately obtained from the dual lemma in [2] by multiplying $s_k$. We then introduce two lemmas to pave the way to the point-wise concentration:

**Lemma 1.2** (Boucheron's Inequality)**.** *Let $h : \mathbb{R}^n \to \mathbb{R}$ be convex or concave and L-Lipschitz with respect to the $\|\cdot\|_2$-norm. Let $\mathbf{Z}_i$ be the independent random variables with $\mathbf{Z}_i \in [a, b]$. Then for $t \geq 0$ and stacked $\mathbf{Z}_i$ as $\mathbf{Z}^n$*

$$\mathbb{P}\left(|h(\mathbf{Z}^n) - \mathbb{E}[h(\mathbf{Z}^n)]| \geq t\right) \leq 2\exp\left(-\frac{t^2}{2L^2(b-a)^2}\right) \tag{4}$$

**Lemma 1.3.** *Let $\mathbf{Z}_i$ be an i.i.d. sequence of random variables with $|\mathbf{Z}_i| \leq M_1$, then the following inequality holds:*

$$\mathbb{E}\left[|\mathbf{Z}|^{k_*}\right]^{\frac{1}{k_*}} - \frac{2}{k}\sqrt{M_1} n_s^{-\frac{1}{\max(k_*, 2)}} \leq \mathbb{E}\left[\left(\frac{1}{n_s}\sum_{i=1}^{n_s} |\mathbf{Z}_i|^{k_*}\right)^{\frac{1}{k_*}}\right] \leq \mathbb{E}\left[|\mathbf{Z}|^{k_*}\right]^{\frac{1}{k_*}} \tag{5}$$

Then we prove the following point-wise concentration theorem that for each parameter $\theta \in \Theta$, $\mathcal{R}_k(\ell(\theta); \widehat{P}_N^S) \xrightarrow{P} \mathcal{R}_k(\ell(\theta); P^S)$:

**Theorem 1.4.** *For a fixed $\theta$ and $u \geq 0$, the following inequality holds with probability $1 - 2e^{-u}$:*

$$\|\mathcal{R}_k(\ell(\theta); \widehat{P}_N^S) - \mathcal{R}_k(\ell(\theta); P^S)\| \leq 5 s_k \rho_k^2 n_s^{-\frac{1}{\max(k_*,2)}} \left(\frac{1}{k} + \sqrt{u + \log n_s}\right)\left(\sqrt{v(\rho_k, M_1)} + \frac{1}{2}\right)^2, \tag{6}$$

*where* $v(\rho_k, M_1) = \max\left(\frac{\rho_k}{\rho_k - 1}, 2\right) M_1$.

*Proof.* Letting $g_k(P^S) = \rho_k(\mathbb{E}_{P^S}[(\ell(f(\mathbf{X}^S \mid \theta), \mathbf{Y}^{err}) - \eta)_+^{k_*}])^{\frac{1}{k_*}} + \eta$ and $g_k(\widehat{P}_N^S) = \rho_k(\frac{1}{n_s}\sum_{i=1}^{n_s} \|\ell(f(\mathbf{X}_i^S \mid \theta), \mathbf{Y}_i^{err}) - \eta\|_i^{k_*})^{\frac{1}{k_*}} + \eta$, we first show that $g_k(\widehat{P}_N^S)$ convergences to $g_k(P^S)$ in probability. Based on previous conclusion, $g_k(\widehat{P}_N^S)$ as the function of the vector $\ell(f(\mathbf{X}_i^S \mid \theta), \mathbf{Y}_i^{err})_{i=1}^{n_s} \in \mathbb{R}^{n_s}$ is $\rho_k n_s^{-\frac{1}{max(2,k_*)}}$-Lipschitz under the $\|\cdot\|_2$ norm (Lemma 7 in [2]). Meanwhile, $g_k(\widehat{P}_N^S)$ is convex with respect to $\eta$ [2] and achieves the infimum when $\eta$ falls in the interval $\left[-\frac{1}{c_k - 1} M_1, M_1\right]$ (Lemma 9 in [2]). We then obtain the the interval of $\mathbf{Z} = (\ell(f(\mathbf{X}^S \mid \theta), \mathbf{Y}^{err}) - \eta)_+$ as $\max(2, \frac{\rho_k}{\rho_k - 1}) M_1$ and apply the Bouncheron's inequality to the function $g_k(\widehat{P}_N^S)(\ell(f(\mathbf{X}_i^S \mid \theta), \mathbf{Y}_i^{err})_{i=1}^{n_s})$ to obtain the following concentration result on $g_k(\widehat{P}_N^S) \xrightarrow{P} \mathbb{E}_{P_S}[g_k(\widehat{P}_N^S)]$:

$$\left|g_k(\widehat{P}_N^S) - \mathbb{E}_{P_S}[g_k(\widehat{P}_N^S)]\right| \leq \sqrt{2u}\rho_k \max\left(\frac{\rho_k}{\rho_k - 1}, 2\right) M_1 n_s^{-\frac{1}{\max(k_*,2)}}, \tag{7}$$

where above inequality (7) holds with the probability as $1 - 2e^{-u}$. Meanwhile, another inequality in Lemma 1.3 bounds the difference between $\mathbb{E}_{P_S}[g_k(\widehat{P}_N^S)]$ and $g_k(P_S)$ as follows:

$$\left|g_k(P_S) - \mathbb{E}_{P_S}[g_k(\widehat{P}_N^S)]\right| \leq \frac{2\rho_k}{k}\sqrt{\max\left(\frac{\rho_k}{\rho_k - 1}, 2\right) M_1 n_s^{-\frac{1}{\max(k_*,2)}}}. \tag{8}$$

Then we obtain the concentration results as $g_k(\widehat{P}_N^S) \xrightarrow{P} g_k(P_S)$ with the following inequality holds with probability $1 - 2e^{-u}$:

$$\left|g_k(\widehat{P}_N^S) - g_k(P_S)\right| \leq \rho_k n_s^{-\frac{1}{\max(k_*,2)}} \left(\frac{2}{k} + \sqrt{2u}\right)\left(\sqrt{v(\rho_k, M_1)} + v(\rho_k, M_1)\right), \tag{9}$$

where $v(\rho_k, M_1) = \max\left(\frac{\rho_k}{\rho_k - 1}, 2\right) M_1$. Meanwhile, we denote the term $\rho_k n_s^{-\frac{1}{\max(k_*,2)}} \left(\frac{2}{k} + \sqrt{2u}\right)\left(\sqrt{v(\rho_k, M_1)} + v(\rho_k, M_1)\right)$ by $\epsilon(u, v, n_s)$ for convenience. Recalling the $1 + \rho_k$-Lipschitzness of $g_k(\widehat{P}_N^S)$ and $g_k(P_S)$ with respect to $\eta$, we divide the interval for $\eta$ into sub-intervals $\eta_i := -(\rho_k - 1)^{-1} M_1 + i\epsilon(u, v, n_s)$ with nonnegative integers $i \leq \frac{\rho_k}{\rho_k - 1}\frac{M_1}{\epsilon(u,v,n_s)}$. Consequently, the point-wise concentration of $\mathcal{R}_k(\ell(\theta); \widehat{P}_N^S) \xrightarrow{P} \mathcal{R}_k(\ell(\theta); P^S)$ can be formulated in the following inequality holds with probability $1 - 2\exp\left(-u + \log\frac{\rho_k}{\rho_k - 1}\frac{M_1}{\epsilon(u,v,n_s)}\right)$:

$$\begin{aligned}
&\left|\mathcal{R}_k(\ell(\theta); \widehat{P}_N^S) - \mathcal{R}_k(\ell(\theta); P^S)\right| = s_k \left|\inf_{\eta} g_k(\widehat{P}_N^S) - \inf_{\eta} g_k(P_S)\right| \\
&= s_k \left|\inf_{\eta \in \left[-\frac{1}{c_k - 1} M_1, M_1\right]} g_k(\widehat{P}_N^S) - \inf_{\eta \in \left[-\frac{1}{c_k - 1} M_1, M_1\right]} g_k(P_S)\right| \\
&\leq s_k \sup_{\eta \in \left[-\frac{1}{c_k - 1} M_1, M_1\right]} \left|g_k(\widehat{P}_N^S) - g_k(P_S)\right| \\
&\leq s_k \sup_{\eta \in \left[-\frac{1}{c_k - 1} M_1, M_1\right]} \left|g_k(\widehat{P}_N^S)(\eta) - g_k(\widehat{P}_N^S)(\eta_i)\right| + \left|g_k(\widehat{P}_N^S)(\eta_i) - g_k(P_S)(\eta_i)\right| + \left|g_k(P_S)(\eta_i) - g_k(P_S)(\eta)\right| \\
&\leq s_k \max_{\eta_i} \left|g_k(\widehat{P}_N^S)(\eta_i) - g_k(P_S)(\eta_i)\right| + 2(1 + \rho_k)\epsilon(u, v, n_s) \\
&\leq s_k(3 + 2\rho_k)\epsilon(u, v, n_s).
\end{aligned} \tag{10}$$

Equivalently, let $u = t + (1 + \frac{1}{\max(k_*,2)}) \log n_s$, we obtain the final point-wise concentration bound as with probability $1 - 2e^{-t}$

holds:

$$\begin{aligned}&\left|\mathcal{R}_k(\ell(\theta);\widehat{P}_N^S)-\mathcal{R}_k(\ell(\theta);P^S)\right|\le s_k(2\rho_k+3)\epsilon(u,v,n_s)\\&\le 5s_k\rho_k\epsilon(u,v,n_s)\\&\le 5s_k\rho_k^2 n_s^{-\frac{1}{\max(k_*,2)}}(\frac{1}{k}+\sqrt{t+\log n_s})(\sqrt{\max\left(\frac{\rho_k}{\rho_k-1},2\right)M_1}+\frac{1}{2})^2\end{aligned}\tag{11}$$

□

We let $\mathcal{F}=\{f(\cdot\mid\theta)\}$ be the model space and equip it with the norm $\|f\|_{\mathcal{L}_\infty}=\sup_x f(x\mid\theta)$. Then we define the $\epsilon$-covering number of $\mathcal{F}$ as $N(\mathcal{F},\xi)$ and assume the compactness of $\mathcal{F}$ such that $\mathcal{F}$ has a finite $\xi$-cover with the covering set consisting of $N(\mathcal{F},\xi)$ points. More specifically, we let $N(\mathcal{F},\frac{\xi(u,v,n_s)}{3})$ with $\xi(u,v,n_s)=\frac{5\rho_k}{C}n_s^{-\frac{1}{\max(k_*,2)}}(\frac{1}{k}+\sqrt{u+\log n_s})(\sqrt{v(\rho_k,M_1)}+\frac{1}{2})^2$ be a $\xi(u,v,n_s)$-finite cover of $\mathcal{F}$ and $V(\mathcal{F},\frac{\xi(u,v,n_s)}{3})$ be the corresponding parameter set for the covering set in $\mathcal{F}$, where we assume that the loss function $\ell(f(\cdot\mid\theta),\cdot)$ is $C$-Lipschitz with respect to its first parameter $f(\cdot\mid\theta)$ with the norm $\|\cdot\|_\infty$. Then we obtain the following theorem of the uniform convergence of DSO over parameter space $\Theta$:

**Theorem 1.5.** *For $u\ge 0$ and the loss function $\ell(f(\cdot\mid\theta),\cdot)$ that is $C$-Lipschitz with respect to its first parameter $f(\cdot\mid\theta)$ with the norm $\|\cdot\|_\infty$, the following inequality holds with probability $1-2N(\mathcal{F},\frac{\xi(u,v,n_s)}{3})e^{-u}$:*

$$\begin{aligned}&\mathcal{R}_k(\ell(\hat{\theta}_{n_s});P^S)-\inf_{\theta\in\Theta}\mathcal{R}_k(\ell(\theta);P^S)\\&\le 30s_k\rho_k^2 n_s^{-\frac{1}{\max(k_*,2)}}(\frac{1}{k}+\sqrt{u+\log n_s})(\sqrt{\max\left(\frac{\rho_k}{\rho_k-1},2\right)M_1}+\frac{1}{2})^2.\end{aligned}\tag{12}$$

*Proof.* In order to bound the term $\mathcal{R}_k(\ell(\theta);\widehat{P}_N^S)-\inf_{\theta\in\Theta}\mathcal{R}_k(\ell(\theta);P^S)$, we first have to bound the supremum of their difference over $\Theta$:

$$\begin{aligned}&\sup_{\theta\in\Theta}\left|\mathcal{R}_k(\ell(\theta);\widehat{P}_N^S)-\mathcal{R}_k(\ell(\theta);P^S)\right|\\&\le\sup_{\theta\in\Theta}\left(\left|\mathcal{R}_k(\ell(\theta);\widehat{P}_N^S)-\mathcal{R}_k(\ell(\theta_i);\widehat{P}_N^S)\right|+\left|\mathcal{R}_k(\ell(\theta);P_S)-\mathcal{R}_k(\ell(\theta_i);P_S)\right|+\left|\mathcal{R}_k(\ell(\theta_i);P_S)-\mathcal{R}_k(\ell(\theta_i);\widehat{P}_N^S)\right|\right)\\&\le\max_{\theta_i\in V(\mathcal{F},\frac{\xi(u,v,n_s)}{3})}\left|\mathcal{R}_k(\ell(\theta_i);P_S)-\mathcal{R}_k(\ell(\theta_i);\widehat{P}_N^S)\right|+\frac{2}{3}s_k\rho_k C\xi(u,v,n_s),\end{aligned}\tag{13}$$

where the second inequality comes from that $\theta_i\in V(\mathcal{F},\frac{\xi(u,v,n_s)}{3})$ provides a $\frac{\xi(u,v,n_s)}{3}$-finite cover of $\mathcal{F}$, the $s_k\rho_k$-Lipschitzness of $\mathcal{R}_k(\ell(\theta_i);\widehat{P}_N^S)$ and $\mathcal{R}_k(\ell(\theta_i);P_S)$ with respect to $\ell$ and the $C$-Lipschitzness of $\ell$ with respect to $f$. Then the union bound provides that:

$$\begin{aligned}&\sup_{\theta\in\Theta}P(\left|\mathcal{R}_k(\ell(\theta);\widehat{P}_N^S)-\mathcal{R}_k(\ell(\theta);P^S)\right|\ge s_k\rho_k C\xi(u,v,n_s))\\&\le N(\mathcal{F},\frac{\xi(u,v,n_s)}{3})\max_{\theta_i\in V}P(\left|\mathcal{R}_k(\ell(\theta_i);P_S)-\mathcal{R}_k(\ell(\theta_i);\widehat{P}_N^S)\right|\ge\frac{1}{3}s_k\rho_k C\xi(u,v,n_s)).\end{aligned}\tag{14}$$

Applying the Theorem 1.4 into the R.H.S in (14), we obtain the following inequality holds with probability at least $1-2N(\mathcal{F},\frac{\xi(u,v,n_s)}{3})e^{-u}$:

$$\sup_{\theta\in\Theta}\left|\mathcal{R}_k(\ell(\theta);\widehat{P}_N^S)-\mathcal{R}_k(\ell(\theta);P^S)\right|\le 3s_k\rho_k C\xi(u,v,n_s).\tag{15}$$

Finally, combined with the fact that $\hat{\theta}_{n_s}$ is the empirical risk minimizer under empirical distribution $\widehat{P}_N^S$, we obtain the uniform bound of the concentration process of DSO over $\Theta$ via the following inequality holds with probability at least

$1-2N(\mathcal{F},\frac{\xi(u,v,n_s)}{3})e^{-u}$:

$$\begin{aligned}
&\mathcal{R}_k(\ell(\hat{\theta}_{n_s});P^S)-\mathcal{R}_k(\ell(\theta);P^S)\\
&=\mathcal{R}_k(\ell(\hat{\theta}_{n_s});P^S)-\mathcal{R}_k(\ell(\hat{\theta}_{n_s});\widehat{P}_N^S)+\mathcal{R}_k(\ell(\hat{\theta}_{n_s});\widehat{P}_N^S)-\mathcal{R}_k(\ell(\theta);P^S)\\
&\leq\mathcal{R}_k(\ell(\hat{\theta}_{n_s});P^S)-\mathcal{R}_k(\ell(\hat{\theta}_{n_s});\widehat{P}_N^S)+\mathcal{R}_k(\ell(\theta);\widehat{P}_N^S)-\mathcal{R}_k(\ell(\theta);P^S)\\
&\leq 6s_k\rho_k C\xi(u,v,n_s)\\
&\leq 30s_k\rho_k^2 {n_s}^{-\frac{1}{\max(k_\star,2)}}(\frac{1}{k}+\sqrt{u+\log n_s})(\sqrt{\max\left(\frac{\rho_k}{\rho_k-1},2\right)M_1}+\frac{1}{2})^2,
\end{aligned}\tag{16}$$

where the we take $\inf$ over $\Theta$ on both sides of the inequality and the proof finishes. □

We then combine the concentration result on supervised objective into Theorem 1.5 with the following lemma:

**Lemma 1.6.** *Let $\ell(f(\cdot\mid\theta),\cdot)$ be $C$-Lipschitz with respect to the first term under $\|\cdot\|_\infty$-norm. Assume the compactness of $\mathcal{F}=\{f(\cdot\mid\theta)\}$ with $\|\cdot\|_\infty$-norm and bounded loss with $\left|\ell(f(\mathbf{X}^S\mid\theta),\mathbf{Y}^S)\right|\leq M_2$. Then the following inequality holds with probability at least $1-2N(\mathcal{F},\sqrt{\frac{2u}{n_s}}\frac{M_2}{4C})e^{-u}$:*

$$P\left(\sup_{\theta\in\Theta}\left|\mathbb{E}_{P_S}\left[\ell(f(\mathbf{X}^S\mid\theta),\mathbf{Y}^S)\right]-\frac{1}{n_s}\sum_{i=1}^{n_s}\ell(f(\mathbf{X}_i^S\mid\theta),\mathbf{Y}_i^S)\right|\right)\leq 2\sqrt{\frac{2u}{n_s}}M_2.\tag{17}$$

*Proof.* We let $N_0=N(\mathcal{F},\sqrt{\frac{2u}{n_s}}\frac{M_2}{4C})$, $L_\theta=\mathbb{E}_{P_S}\left[\ell(f(\mathbf{X}^S\mid\theta),\mathbf{Y}^S)\right]$ and $\widehat{L}_\theta=\frac{1}{n_s}\sum_{i=1}^{n_s}\ell(f(\mathbf{X}_i^S\mid\theta),\mathbf{Y}_i^S)$. Meanwhile, the compactness assumption of $\mathcal{F}$ renders $\mathcal{F}$ to be totally bounded, such that $\mathcal{F}$ is covered by the finite set $\{B(f_i,\sqrt{\frac{2u}{n_s}}\frac{M_2}{4C})\}_{i=1}^{N_0}$, where $B(f_i,\sqrt{\frac{2u}{n_s}}\frac{M_2}{4C})$ refers to the open ball centered at $f_i$ with radius $\sqrt{\frac{2u}{n_s}}\frac{M_2}{4C}$. Then the following inequality holds:

$$\begin{aligned}
P\left(\sup_{\theta\in\Theta}\left|L_\theta-\widehat{L}_\theta\right|\geq\sqrt{\frac{2u}{n_s}}M_2\right)&\leq\sum_{i=1}^{N_0}P\left(\sup_{\theta\in B(f_i)}\left|L_\theta-\widehat{L}_\theta\right|\geq\sqrt{\frac{2u}{n_s}}M_2\right)\\
&\leq\sum_{i=1}^{N_0}P\left(\left|L_{f_i}-\widehat{L}_{f_i}\right|\geq\sqrt{\frac{u}{2n_s}}M_2\right),
\end{aligned}\tag{18}$$

where the first inequality is due to the union bound and the second one is due to the triangle inequality with the fact that $\left|\left|L_{f_i}-\widehat{L}_{f_i}\right|-\left|L_\theta-\widehat{L}_\theta\right|\right|\leq 2C\|f_i-\theta\|_\infty\leq\sqrt{\frac{u}{2n_s}}M_2$. Then we apply the Hoeffding's inequality on the random variables $L_{f_i}-\widehat{L}_{f_i}$ and obtain the final bound. □

To distinguish from the DSO objective $\mathcal{R}_k(\ell(\hat{\theta}_{n_s});P^S)$ and $\mathcal{R}_k(\ell(\theta);P_S$, we denote the expected and empirical versions of the total objective by $\mathcal{R}_{Total}$ and $\widehat{\mathcal{R}}_{Total}^N$, respectively:

**Theorem 1.7.** *For $u\geq 0$, the loss function $\ell(f(\cdot\mid\theta),\cdot)$ that is $C$-Lipschitz with respect to its first parameter $f(\cdot\mid\theta)$ and the fact that both $\left|\ell(f(\mathbf{X}^S\mid\theta),\mathbf{Y}^{err})\right|\leq M_1$ and $\left|\ell(f(\mathbf{X}^S\mid\theta),\mathbf{Y}^S)\right|\leq M_2$ are bounded, the following inequality holds with probability $1-2\left(N(\mathcal{F},\frac{\xi(u,v,n_s)}{3})+N(\mathcal{F},\sqrt{\frac{2u}{n_s}}\frac{M_2}{4C})\right)e^{-u}$:*

$$\widehat{\mathcal{R}}_{Total}^N-\inf_{\theta\in\Theta}\mathcal{R}_{Total}\leq 2\sqrt{\frac{2u}{n_s}}M_2+6s_k\rho_k\xi(u,v,n_s),\tag{19}$$

*where $\xi(u,v,n_s)=5\rho_k{n_s}^{-\frac{1}{\max(k_\star,2)}}(\frac{1}{k}+\sqrt{u+\log n_s})(\sqrt{v(\rho_k,M_1)}+\frac{1}{2})^2$.*

*Proof.* It is directly obtained from Theorem (1.5) and Lemma (1.6). □

### 1.3. Authorization analysis of DSO

**Lemma 1.8.** *Assume that $P$ and $P_U$ have the same support and $|\ell(f(\mathbf{X}), \mathbf{Y}^{err})| \leq M_1$, then $\mathbb{E}_{P_U}\left[\ell(f(\mathbf{X}), \mathbf{Y}^{err})\right]$ is bounded by $\mathbb{E}_P\left[\ell(f(\mathbf{X}), \mathbf{Y}^{err})\right]$ with the F-divergence $D_{\phi_k}(P_U \| P)$ as follows:*

$$\mathbb{E}_{P_U}\left[\ell(f(\mathbf{X}), \mathbf{Y}^{err})\right] \leq k(k-1) D_{\phi_k}(P_U \| P)^{\frac{1}{k}} M_1^{\frac{1}{k}} \mathbb{E}_P\left[\ell(f(\mathbf{X}), \mathbf{Y}^{err})\right]^{1-\frac{1}{k}} \tag{20}$$

*Proof.* Due to the covariate assumption between $P_U$ and $P$, we reformulate $\mathbb{E}_{P_U}\left[\ell(f(\mathbf{X}), \mathbf{Y}^{err})\right]$ as follows:

$$\begin{aligned}
\mathbb{E}_{P_U}\left[\ell(f(\mathbf{X}), \mathbf{Y}^{err})\right] &= \mathbb{E}_P\left[\frac{P_U(\mathbf{X})}{P(\mathbf{X})}\ell(f(\mathbf{X}), \mathbf{Y}^{err})\right] \\
&= \int \left[P(\mathbf{X})^{\frac{1}{k}} \frac{P_U(\mathbf{X})}{P(\mathbf{X})} P(\mathbf{X})^{\frac{k-1}{k}} \ell(f(\mathbf{X}), \mathbf{Y}^{err})\right] d\mathbf{x} \\
&\leq \left[\int P(\mathbf{X}) \left(\frac{P_U(\mathbf{X})}{P(\mathbf{X})}\right)^k\right]^{\frac{1}{k}} \left[\int P(\mathbf{X}) (\ell(f(\mathbf{X}), \mathbf{Y}^{err}))^{\frac{k}{k-1}}\right]^{\frac{k-1}{k}} \\
&\leq k(k-1) D_{\phi_k}(P_U \| P)^{\frac{1}{k}} M_1^{\frac{1}{k}} \mathbb{E}_P\left[\ell(f(\mathbf{X}), \mathbf{Y}^{err})\right]^{1-\frac{1}{k}},
\end{aligned} \tag{21}$$

where we apply the Holder's inequality in the third inequality. □

## 2. Implementation Details

### 2.1. Connections between DPA baselines and OOD detection

As we mentioned in the main part, one can also consider the problem of SDPA from the perspective of OOD detection. To be specific, the authorizer could, for example, use any OOD detection method and simply return a wrong label for OOD samples. Among the wide range of OOD detection areas [11], the branch with relevance to our problem is "Near OOD Detection", which is recently proposed to detect OOD samples that distribute close to the training one [3]. However, we observe that it is unnecessary to such baselines in addition. The reason is simple: **the framework of such Near OOD detection methods is almost equivalent to the DPA baseline GNTL we have already compared.**

To make clear comparisons, we list the recently proposed SOTA variant of near OOD detection [6] and GNTL [10] as follows, respectively:

(a) [6] builds up a generative model (e.g., diffusion model) and simulates the OOD samples by manually tuning the parameters of the generative model. Then the model for deployment is trained to distinguish the generated samples (OOD) from the training one.

(b) GNTL [10] first builds up a GAN model and then generates several synthetic target domains by tuning the GAN. Finally, the classifier for deployment is trained to discriminate the training samples from generated ones.

One can clearly observe that the above two pipelines coincide. Moreover, such first-generate-then-discriminate approaches suffer from two problems:

(a) Can the generated target domains be representative enough? In other words, can the authorizer tune the GAN models to generate close distributions in every direction over the divergence ball? Unfortunately, the answer is no. Geometrically, such methods generate target data only in several directions with some distances in the distributional divergence ball. The number of generated target domains is controlled manually by tuning the generative models. However, near distributions in the ball are infinite such that most near distributions will be definitely neglected.

(b) Moreover, generation-based solutions are very inconvenient. One has to tune and re-train the underlying GAN/diffusion models again once new authorized/training data comes. Such inflexible approaches further prevent their realistic applications.

Finally, as shown in above, GNTL can be equivalently regarded as a near OOD detection approach. Therefore, we argue that comparison with more OOD detection solutions is meaningless.

### 2.2. Details on Datasets

Among the six benchmarks, we resize the images from Digits and Cifar10 & STL10 into $32 \times 32$, while the images from the rest benchmarks are resizes into $224 \times 224$. Following [8], we have removed the non-overlapping classes ("frog" and "monkey") and reduce the problem to a nine-class classification problem for Cifar10 & STL10. For all datasets, we take their own training sets as the source data, and their own testing sets as the test data.

### 2.3. Details on the Benchmark Solution

In this section, we provide detailed settings of GNTL (the compared benchmark solution). We conduct all the experiments using the original implementation of [10] by Pytorch. For benchmarks including Digits and Cifar10 & STL10, we directly utilizes the GAN architecture designed in [10] without any modification. For the rest benchmarks with resized images as $224 \times 224$, we first resize them into $112 \times 112$ (to follow the original implementation in [10] on the Visda benchmark) and uses the same GAN model as the one designed for Visda dataset in [10]. Moreover, unlike [10], we apply GNTL to source-only deployment authorization without embedding any pre-defined watermarks into the training data. For image data, the pre-defined watermarks is usually achieved by attaching some shallow patches into the images [10]. In fact, the source-only deployment authorization in [10] is governed by the ownership authorization, such that the training data is embedded with patches by default. However, we aim to develop solutions for the pure deployment authorization problem regardless of other kinds of authorization. For instance, consider the online services of machine learning models, users cannot access the weights of the networks such that the ownership problem does not exist. On the contrary, the augmentation strategy with patch embedding in [10] might ruin the performance of the model when users upload images without any watermark. Therefore, we apply GNTL to each task with two stages as: (a) perform GAN augmentation on the training domain and obtain the augmented data as the auxiliary domain [10]; (b) apply the standard NTL method on the training domain and the generated auxiliary domain. For the optimization of GNTL, we strictly follow the original settings in [10] to tune the GAN augmentation and the NTL training process. More specifically, we utilize Adam [5] as the optimizer to implement the NTL training process, with the initial learning rate set as $\gamma = 0.0001$ and the batch size set as 32. In the training of GAN augmentation, the optimizer is also Adam, while the initial learning rate is set as $\gamma = 0.0002$ with two decay momentums set as 0.5 and 0.999. The batch size is 64, and the dimension of the latent space fed to the generator is 256 [10].

### 2.4. Details on Our Methods

To support the classification tasks, we follow the previous protocols [10] and apply the VGG-11 network [9] on the Digit benchmark, while the rest benchmarks are trained and tested with the resnet-50 model [4]. To imitate the real-world case, all the models are pre-trained on the Imagenet dataset [10]. For both DSO and TDSO, we utilize Adam [5] as the optimizer, while the initial learning rate is set as $\gamma = 0.00005$ with the batch size set as 32. Regarding our implementation, we upload a Pytorch example of our code in the appendix, while the cleaned version of the total project will be released once the paper is accepted.

## 3. Experiments

In this section, we report detailed results of the source-only deployment authorization and target-combined authorization on six benchmarks. Tab. 1, Tab. 4, Tab. 2 and Tab. 6 report the training accuracy and the unauthorized performance drop of each solution for each task (e.g., Amazon → Webcam in Office-31) on the Office-31, Digit, Visda and Cifar10 & STL10 benchmarks. Tab. 3 and Tab. 5 report the corresponding results on PACS and VLCS benchmarks, respectively. Note that Tab. 3 and Tab. 5 also report the authorization performance when training each solution on the combination of the three domains, while the other domain is set as the unauthorized domain. For instance, we select each domain in PACS as the unauthorized domain (e.g., $S$), while the other three domains are set as training data (e.g., A + C + P). Such a setting corresponds to the real-world case that the training data consists of multiple domains.

Table 1: Classification accuracy on Office-31 benchmark including Amazon (A), Webcam (W) and DSLR (D) for source-only and target-combined deployment authorization tasks, where the bold value represents the best performance for each authorization scene (source-only and target-combined) in each row. **Notably, the source domain refers to the authorized domain, and the other domains are considered as unauthorized domains. Meanwhile, target domains contain both the source domain and unauthorized domains.** Moreover, we denote the training performance and unauthorized performance for TDSO in the form of Training_acc ⇒ Target_acc (e.g., $0.99 \Rightarrow 0.05$ represents that TDSP preserves the training accuracy as $99\%$ with the unauthorized accuracy as $5\%$).

| Task | | Supvised | Source-Only | | Target-Combined |
|---|---|---|---|---|---|
| Methods | | Resnet-50 | G-NTL | DSO | TDSO |
| Source Domain | Target Domains | Performance | | | |
| Amazon | Amazon | 0.99 | 0.99 | 0.98 | |
| | Webcam | 0.70 | 0.62 | 0.39 | 0.99 ⇒ 0.05 |
| | DSLR | 0.69 | 0.64 | 0.45 | 0.99 ⇒ 0.04 |
| Drop on Authorized Data | | 0.0% | 0.0% | 1.0% | 0.0% |
| Drop on Unauthorized Data | | 0.0% | 8.5% | **27.5**% | **64.5**% |
| Webcam | Amazon | 0.64 | 0.59 | 0.20 | 0.99 ⇒ 0.04 |
| | Webcam | 0.99 | 0.99 | 0.96 | |
| | DSLR | 0.96 | 0.93 | 0.46 | 0.99 ⇒ 0.03 |
| Drop on Authorized Data | | 0.0% | 0.0% | 3.0% | 0.0% |
| Drop on Unauthorized Data | | 0.0% | 5.5% | **47.0**% | **76.5**% |
| DSLR | Amazon | 0.63 | 0.44 | 0.20 | 0.99 ⇒ 0.09 |
| | Webcam | 0.96 | 0.89 | 0.41 | 0.99 ⇒ 0.05 |
| | DSLR | 0.99 | 0.96 | 0.96 | |
| Drop on Authorized Data | | 0.0% | 3.0% | 3.0% | 0.0% |
| Drop on Unauthorized Data | | 0.0% | 13.0% | **49.0**% | **72.5**% |

Table 2: Classification accuracy on Visda2017 for source-only and target-combined deployment authorization tasks, where the bold value represents the best performance for each authorization scene (source-only and target-combined) in each row. **Notably, the source domain refers to the authorized domain, and the other domains are considered as unauthorized domains. Meanwhile, target domains contain both the source domain and unauthorized domains.**

| Task | | Supvised | Source-Only | | Target-Combined |
|---|---|---|---|---|---|
| Methods | | Resnet-50 | G-NTL | DSO | TDSO |
| Source Domain | Target Domains | Performance | | | |
| Real | Real | 0.94 | 0.89 | 0.94 | |
| | Synthetic | 0.80 | 0.74 | 0.52 | 0.94 ⇒ 0.08 |
| Drop on Authorized Data | | 0.0% | 5.0% | 0.0% | 0.0% |
| Drop on Unauthorized Data | | 0.0% | 6.0% | **18.0**% | **72.0**% |
| Synthetic | Real | 0.35 | 0.34 | 0.19 | 0.95 ⇒ 0.07 |
| | Synthetic | 0.95 | 0.95 | 0.94 | |
| Drop on Authorized Data | | 0.0% | 0.0% | 1.0% | 0.0% |
| Drop on Unauthorized Data | | 0.0% | 1.0% | **16.0**% | **28.0**% |

Table 3: Classification accuracy on PACS benchmark including Art_paintint (A), Cartoon (C), Photo (P) and Sketch (S) for source-only and target-combined deployment authorization tasks, where the bold value represents the best performance for each authorization scene (source-only and target-combined) in each row. **Notably, the source domain refers to the authorized domain, and the other domains are considered as unauthorized domains. Meanwhile, target domains contain both the source domain and unauthorized domains.**

| Task | | Supvised | Source-Only | | Target-Combined |
|---|---|---|---|---|---|
| Methods | | Resnet-50 | G-NTL | DSO | TDSO |
| Source Domain | Target Domains | Performance | | | |
| Art | Art | 0.99 | 0.99 | 0.98 | |
| | Cartoon | 0.58 | 0.55 | 0.53 | 0.99⇒0.15 |
| | Photo | 0.78 | 0.75 | 0.51 | 1.00⇒0.11 |
| | Sketch | 0.55 | 0.54 | 0.45 | 1.00⇒0.12 |
| Drop on Authorized Data | | 0.0% | 0.0% | 1.0% | 0% |
| Drop on Unauthorized Data | | 0.0% | 2.0% | **13.0**% | **51.0**% |
| Cartoon | Art | 0.60 | 0.59 | 0.49 | 0.99⇒0.13 |
| | Cartoon | 1.00 | 0.97 | 0.99 | |
| | Photo | 0.63 | 0.63 | 0.55 | 0.99⇒0.15 |
| | Sketch | 0.61 | 0.61 | 0.54 | 0.99⇒0.14 |
| Drop on Authorized Data | | 0.0% | 3.0% | 1.0% | 1.0% |
| Drop on Unauthorized Data | | 0.0% | 0.3% | **8.0**% | **47.3**% |
| Photo | Art | 0.57 | 0.59 | 0.49 | 0.99⇒0.15 |
| | Cartoon | 0.50 | 0.55 | 0.29 | 0.99⇒0.09 |
| | Photo | 1.00 | 0.99 | 0.99 | |
| | Sketch | 0.52 | 0.22 | 0.21 | 0.99⇒0.15 |
| Drop on Authorized Data | | 0.0% | 1.0% | 1.0% | 1.0% |
| Drop on Unauthorized Data | | 0.0% | 9.7% | **20.0**% | **40.0**% |
| Sketch | Art | 0.24 | 0.21 | 0.13 | 0.99⇒0.09 |
| | Cartoon | 0.38 | 0.37 | 0.22 | 0.99⇒0.12 |
| | Photo | 0.22 | 0.20 | 0.14 | 0.99⇒0.18 |
| | Sketch | 0.99 | 0.99 | 0.98 | |
| Drop on Authorized Data | | 0.0% | 0.0% | 1.0% | 0.0% |
| Drop on Unauthorized Data | | 0.0% | 2.0% | **12.0**% | **15.0**% |
| A+C+P → S | | 0.99⇒0.80 | 0.99⇒0.64 | 0.95⇒0.57 | 0.99⇒0.19 |
| C+P+S → A | | 0.99⇒0.81 | 0.99⇒0.66 | 0.95⇒0.47 | 0.98⇒0.23 |
| P+S+A → C | | 0.99⇒0.78 | 0.99⇒0.73 | 0.97⇒0.55 | 0.99⇒0.37 |
| S+A+C → P | | 0.99⇒0.94 | 0.99⇒0.83 | 0.95⇒0.71 | 0.99⇒0.22 |
| Drop on Authorized Data | | 0.0% | 0.0% | 3.5% | 0.0% |
| Drop on Unauthorized Data | | 0.0% | 11.8% | **25.8**% | **58.0**% |

Table 4: Classification accuracy on Digit benchmark including MNIST (MN), USPS (US), SVHN (SV), SYN_D (SD) and MNIST_M(MM) for source-only and target-combined deployment authorization tasks, where the bold value represents the best performance for each authorization scene (source-only and target-combined) in each row. **Notably, the source domain refers to the authorized domain, and the other domains are considered as unauthorized domains. Meanwhile, target domains contain both the source domain and unauthorized domains.**

| Task | | Supvised | Source-Only | | Target-Combined |
|---|---|---|---|---|---|
| Methods | | VGG-11 | G-NTL | DSO | TDSO |
| Source Domain | Target Domains | Performance | | | |
| MN | MN | 0.99 | 0.99 | 0.98 | |
| | US | 0.86 | 0.71 | 0.68 | 0.99⇒ 0.07 |
| | SV | 0.35 | 0.21 | 0.21 | 0.99⇒ 0.08 |
| | SD | 0.36 | 0.22 | 0.11 | 0.98⇒ 0.09 |
| | MM | 0.58 | 0.35 | 0.29 | 0.99⇒ 0.09 |
| Drop on Authorized Data | | 0.0% | 0.0% | 1.0% | 0.0% |
| Drop on Unauthorized Data | | 0.0% | 15.5% | **19.5**% | **45.5**% |
| US | MN | 0.89 | 0.72 | 0.53 | 0.98⇒ 0.09 |
| | US | 0.98 | 0.95 | 0.96 | |
| | SV | 0.26 | 0.14 | 0.12 | 0.98⇒ 0.07 |
| | SD | 0.39 | 0.13 | 0.20 | 0.96⇒ 0.09 |
| | MM | 0.43 | 0.10 | 0.12 | 0.97⇒ 0.10 |
| Drop on Authorized Data | | 0.0% | 3.0% | 2.0% | 0.1% |
| Drop on Unauthorized Data | | 0.0% | 22.0% | **25.0**% | **40.5**% |
| SV | MN | 0.78 | 0.66 | 0.56 | 0.94⇒ 0.10 |
| | US | 0.73 | 0.68 | 0.50 | 0.94⇒ 0.07 |
| | SV | 0.94 | 0.89 | 0.92 | |
| | SD | 0.58 | 0.52 | 0.53 | 0.94⇒ 0.10 |
| | MM | 0.56 | 0.52 | 0.35 | 0.94⇒ 0.09 |
| Drop on Authorized Data | | 0% | 5% | 2% | 0% |
| Drop on Unauthorized Data | | 0.0% | 6.8% | **17.8**% | **57.3**% |
| SD | MN | 0.88 | 0.84 | 0.68 | 0.97⇒ 0.07 |
| | US | 0.83 | 0.81 | 0.69 | 0.97⇒ 0.08 |
| | SV | 0.85 | 0.86 | 0.61 | 0.97⇒ 0.10 |
| | SD | 0.98 | 0.98 | 0.97 | |
| | MM | 0.66 | 0.56 | 0.40 | 0.97⇒ 0.01 |
| Drop on Authorized Data | | 0.0% | 0.0% | 1.0% | 1.0% |
| Drop on Unauthorized Data | | 0.0% | 3.8% | **21.0**% | **74.0**% |
| MM | MN | 0.96 | 0.53 | 0.46 | 0.98⇒ 0.09 |
| | US | 0.83 | 0.13 | 0.19 | 0.98⇒ 0.09 |
| | SV | 0.48 | 0.27 | 0.29 | 0.98⇒ 0.09 |
| | SD | 0.54 | 0.45 | 0.47 | 0.98⇒ 0.10 |
| | MM | 0.98 | 0.97 | 0.94 | |
| Drop on Authorized Data | | 0.0% | 1.0% | 3.0% | 0.0% |
| Drop on Unauthorized Data | | 0.0% | **35.7**% | 35.0% | **61.0**% |

Table 5: Classification accuracy on VLCS benchmark including Caltech101 (C), LabelMe (L), VOC2007 (V) and SUN09 (S) for source-only and target-combined deployment authorization tasks, where the bold value represents the best performance for each authorization scene (source-only and target-combined) in each row. **Notably, the source domain refers to the authorized domain, and the other domains are considered as unauthorized domains. Meanwhile, target domains contain both the source domain and unauthorized domains.**

| Task | | Supvised | Source-Only | | Target-Combined |
|---|---|---|---|---|---|
| Methods | | Resnet-50 | G-NTL | DSO | TDSO |
| Source Domain | Target Domains | Performance | | | |
| Caltech101 | Caltech101 | 1.00 | 0.97 | 1.00 | |
| | LabelMe | 0.28 | 0.26 | 0.25 | 0.97⇒0.22 |
| | VOC2007 | 0.31 | 0.30 | 0.27 | 0.97⇒0.20 |
| | SUN09 | 0.33 | 0.31 | 0.20 | 0.97⇒0.18 |
| Drop on Authorized Data | | 0.0% | 1.7% | 0.0% | 3.0% |
| Drop on Unauthorized Data | | 0.0% | 0.7% | **6.7**% | **10.7**% |
| LabelMe | Caltech101 | 0.83 | 0.77 | 0.33 | 0.99⇒0.23 |
| | LabelMe | 1.00 | 1.00 | 1.00 | |
| | VOC2007 | 0.42 | 0.38 | 0.39 | 0.99⇒0.21 |
| | SUN09 | 0.38 | 0.41 | 0.36 | 0.99⇒0.22 |
| Drop on Authorized Data | | 0.0% | 0.0% | 0.0% | 1.0% |
| Drop on Unauthorized Data | | 0.0% | 2.3% | **18.3**% | **32.3**% |
| VOC2007 | Caltech101 | 0.91 | 0.89 | 0.30 | 0.99⇒0.20 |
| | LabelMe | 0.51 | 0.53 | 0.47 | 0.99⇒0.21 |
| | VOC2007 | 0.99 | 0.99 | 0.96 | |
| | SUN09 | 0.54 | 0.54 | 0.40 | 0.99⇒0.15 |
| Drop on Authorized Data | | 0.0% | 0.0% | 3.0% | 0.0% |
| Drop on Unauthorized Data | | 0.0% | 0.0% | **26.3**% | **46.7**% |
| SUN09 | Caltech101 | 0.59 | 0.59 | 0.33 | 0.98⇒0.21 |
| | LabelMe | 0.52 | 0.51 | 0.49 | 0.98⇒0.20 |
| | VOC2007 | 0.51 | 0.51 | 0.39 | 0.98⇒0.22 |
| | SUN09 | 0.99 | 0.99 | 0.98 | |
| Drop on Authorized Data | | 0.0% | 0.0% | 1.0% | 2.0% |
| Drop on Unauthorized Data | | 0.0% | 0.0% | **13.6**% | **33.3**% |
| C+L+V → S | | 0.99⇒0.70 | 0.99⇒0.62 | 0.96⇒0.46 | 0.97⇒0.22 |
| L+V+S → C | | 0.99⇒0.92 | 0.98⇒0.92 | 0.94⇒0.43 | 0.99⇒0.17 |
| V+S+C → L | | 0.98⇒0.62 | 0.99⇒0.59 | 0.98⇒0.51 | 0.98⇒0.23 |
| S+C+L → V | | 0.99⇒0.71 | 0.99⇒0.61 | 0.99⇒0.42 | 0.98⇒0.21 |
| Drop on Authorized Data | | 0.0% | 0.0% | 2.0% | 0.7% |
| Drop on Unauthorized Data | | 0.0% | 0.0% | **23.0**% | **47.8**% |

Table 6: Classification accuracy on Cifar10 & STL10 for source-only and target-combined deployment authorization tasks, where the bold value represents the best performance for each authorization scene (source-only and target-combined) in each row. **Notably, the source domain refers to the authorized domain, and the other domains are considered as unauthorized domains. Meanwhile, target domains contain both the source domain and unauthorized domains.**

| Task | | Supvised | Source-Only | | Target-Combined |
|---|---|---|---|---|---|
| Methods | | Resnet-50 | G-NTL | DSO | TDSO |
| Source Domain | Target Domains | Performance | | | |
| Cifar10 | Cifar10 | 0.93 | 0.86 | 0.89 | |
| | STL10 | 0.76 | 0.70 | 0.59 | 0.93⇒0.39 |
| Drop on Authorized Data | | 0.0% | 0.0% | 3% | 0.0% |
| Drop on Unauthorized Data | | 0.0% | 6.0% | **17.0**% | **37.0**% |
| STL10 | Cifar10 | 0.68 | 0.57 | 0.43 | 0.92⇒0.17 |
| | STL10 | 0.92 | 0.89 | 0.90 | |
| Drop on Authorized Data | | 0.0% | 0.0% | 1.0% | 2.0% |
| Drop on Unauthorized Data | | 0.0% | 11% | **25.0**% | **51.0**% |

## References


[1] John Duchi, Peter Glynn, and Hongseok Namkoong. Statistics of robust optimization: A generalized empirical likelihood approach. *arXiv preprint arXiv:1610.03425*, 2016.

[2] John Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *arXiv preprint arXiv:1810.08750*, 2018.

[3] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. *Advances in Neural Information Processing Systems*, 34:7068–7081, 2021.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[6] Hossein Mirzaei, Mohammadreza Salehi, Sajjad Shahabi, Efstratios Gavves, Cees GM Snoek, Mohammad Sabokrou, and Mohammad Hossein Rohban. Fake it till you make it: Near-distribution novelty detection by score-based generative models. *arXiv preprint arXiv:2205.14297*, 2022.

[7] Alexander Shapiro. Distributionally robust stochastic programming. *SIAM Journal on Optimization*, 27(4):2258–2275, 2017.

[8] Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. A dirt-t approach to unsupervised domain adaptation. *arXiv preprint arXiv:1802.08735*, 2018.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[10] Lixu Wang, Shichao Xu, Ruiqi Xu, Xiao Wang, and Qi Zhu. Non-transferable learning: A new approach for model ownership verification and applicability authorization. In *International Conference on Learning Representations*, 2021.

[11] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.