Dual Domain Control via Active Learning for Remote Sensing Domain Incremental Object Detection

Supplementary Material

1. Optimal Transmission Algorithm

We consider the optimal transport problem with entropy regularization. Specifically, given two probability distributions μ and ν, we aim to find a joint distribution γ that minimizes the transportation cost from μ to ν, while adding an entropy regularization term to encourage a more uniform and smooth transport plan.

The standard problem of optimal transport is:

$$\min_{\gamma \in \Gamma(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y) d\gamma(x,y), \tag{1}$$

where the set $\Gamma(\mu,\nu)$ is the set of all joint distributions γ that satisfy the marginal conditions, and $c(x,y)$ is the transport cost.

The goal of an optimal transmission problem with entropy regularization is:

$$\min_{\gamma \in \Gamma(\mu,\nu)} \left(\int_{\mathcal{X}\times\mathcal{Y}} c(x,y) d\gamma(x,y) - \epsilon H(\gamma) \right), \tag{2}$$

$$H(\gamma) = -\int_{\mathcal{X}\times\mathcal{Y}} \gamma(x,y) \log \gamma(x,y) d(x,y), \tag{3}$$

where $H(\gamma)$ is the entropy term, which aims to prevent γ from being too concentrated on specific points, thereby encouraging a "smooth" transportation plan. ϵ is a hyperparameter that controls the strength of the regularization and adjusts the weight of the entropy term.

First, we divide the objective function of the optimal transmission problem into two parts: the transportation cost part and the entropy regularization part. To deal with this problem, we introduce the entropy term into the optimization problem and simplify the problem by logarithmic transformation.

Given the cost matrix $C = [c(x_i, y_j)]$, where x_i and y_j are the discretized sample points, and μ and ν are the marginal distributions of the two distributions, we aim to minimize the following objective by solving for the joint distribution γ:

$$F(\gamma) = \sum_{i,j} c_{ij}\gamma_{ij} - \epsilon \sum_{i,j} \gamma_{ij} \log \gamma_{ij}. \tag{4}$$

To facilitate the solution, we introduce Lagrange multipliers to rewrite the marginal constraints (i.e., $\sum_j \gamma_{ij} = \mu_i$, $\sum_i \gamma_{ij} = \nu_j$) as a dual problem. The Lagrange dual function is:

$$\begin{aligned} L(\gamma,u,v) = &\sum_{i,j} c_{ij}\gamma_{ij} - \epsilon \sum_{i,j} \gamma_{ij} \log \gamma_{ij} \\ &+ \sum_i u_i \left(\sum_j \gamma_{ij} - \mu_i \right) \\ &+ \sum_j v_j \left(\sum_i \gamma_{ij} - \nu_j \right), \end{aligned} \tag{5}$$

where u_i and v_j are the Lagrange multipliers, used to handle the marginal constraints.

Taking the derivative of γ_{ij} and setting it to zero, we obtain the following optimization condition:

$$\frac{\partial L}{\partial \gamma_{ij}} = c_{ij} - \epsilon \log \gamma_{ij} - u_i - v_j = 0, \tag{6}$$

solved:

$$\gamma_{ij} = \exp\left(\frac{u_i + v_j - c_{ij}}{\epsilon} \right). \tag{7}$$

Substituting the expression of γ_{ij} into the dual problem, we solve for u_i and v_j. The update rules for u_i and v_j are:

$$u_i = \log\left(\frac{\mu_i}{\sum_j \exp\left(\frac{u_i+v_j-c_{ij}}{\epsilon}\right)} \right), \tag{8}$$

$$v_j = \log\left(\frac{\nu_j}{\sum_i \exp\left(\frac{u_i+v_j-c_{ij}}{\epsilon}\right)} \right). \tag{9}$$

From the above derivation, we obtain the main steps of the Sinkhorn algorithm: Initialize u_i and v_j; Alternately update u_i and v_j until convergence; Update γ_{ij} using the updated u_i and v_j. Repeat the preceding steps until convergence.

The update of u_i and v_j involves summation over all i and j, with a complexity of $O(n^2)$, where n is the number of samples. Therefore, the time complexity of a single iteration is $O(n^2)$.

We referred to the methods of OTDA [1], MOT [3] and OTA [2] in our algorithm design.

2. Experiments and Details

Joint Training. **Sequence I** shows the highest mAP (74.3) and AP50 (97.8) on HRRSD, followed by good performance on the other datasets. However, performance

	mAP	AP_{50}	mAP	AP_{50}	mAP	AP_{50}	mAP	AP_{50}	mAP	AP_{50}	AP_{75}
Sequence I	HRRSD		LEVIR		DIOR		DOTA		Joint Test		
Joint Training	74.3	97.8	61.7	88.6	48.7	77.0	36.3	62.1	44.4	71.2	47.3
Fine-tuning	36.6	71.5	22.1	50.0	26.8	59.5	32.6	61.0	31.3	60.3	30.0
ours	**49.5**	**86.9**	**29.1**	**63.9**	**32.6**	**69.5**	**35.9**	**62.8**	**36.2**	**67.8**	**35.0**
Sequence II	LEVIR		HRRSD		DOTA		DIOR		Joint Test		
Fine-tuning	29.8	58.4	46.6	79.3	19.6	47.4	42.5	71.7	24.5	58.4	14.4
ours	**33.5**	**63.4**	**57.8**	**91.5**	**22.7**	**51.7**	**45.9**	**73.9**	**32.7**	**62.8**	**30.8**
Sequence III	DIOR		DOTA		HRRSD		LEVIR		Joint Test		
Fine-tuning	16.4	41.0	15.3	33.6	36.3	58.4	56.4	76.5	14.8	36.6	8.2
ours	**19.2**	**48.8**	**19.1**	**40.5**	**43.4**	**74.0**	**59.5**	**85.0**	**21.0**	**42.5**	**18.2**

Table 1. Detailed data of sequential fine-tuning experiments.

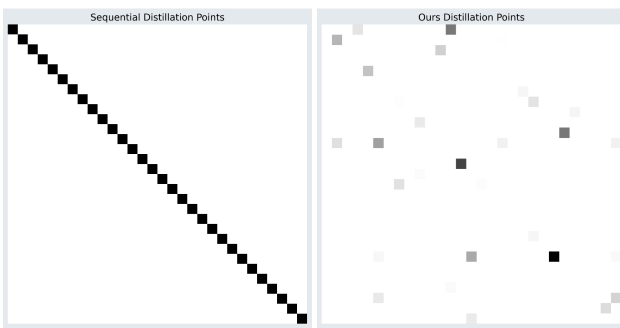

Figure 1. Visual comparison of distillation points for partial queries: sequential distillation and our method.

drops on DOTA (36.3) and DIOR (48.7). **Sequence II** and **III** both exhibit lower mAP values across all datasets, with sequence II having particularly poor performance on the Joint Test (24.5). The *Joint Training* approach, which uses a more traditional training method, leads to stronger performance in initial tasks but suffers from catastrophic forgetting when transferred to new tasks.

Fine-Tuning. Fine-tuning (without our approach) generally shows a substantial drop in performance, especially on earlier tasks, with a notable decline in mAP and AP50, such as on HRRSD and LEVIR. For example, on **Sequence I**, fine-tuning leads to only 36.6 mAP and 71.5 AP50 on HRRSD, while the AP50 drops drastically on DIOR and DOTA (26.8, 32.6). This shows that fine-tuning struggles to maintain performance across domains and shifts, especially when transferring between datasets with differing characteristics.

Ours. In **Sequence I**, our approach significantly outperforms both Joint Training and Fine-tuning methods, with an mAP of 49.5 (up from 36.6 and 74.3, respectively) and an AP50 of 86.9 (up from 71.5 and 97.8). Similarly, in **Sequence II**, we observe strong performance, with mAP values of 33.5 (up from 29.8 in fine-tuning), and the improvement is visible across other datasets. In **Sequence III**, our method again shows the best performance with mAP values of 19.2, compared to 16.4 in fine-tuning. This trend is seen across all sequences, indicating that our approach leads to a more balanced transfer and a better overall performance across different domain shifts.

3. Visualization

We have created a visual representation of the distillation points, as shown in Figure 1. If the fixed weights of a one-to-one query vector are distilled based solely on the index, knowledge transfer can be flawed due to query mismatches. Our ADSC method, however, employs a more effective matching strategy and a many-to-many dynamic weight matching criterion, enabling more efficient knowledge transfer, transmitting richer semantic information, and alleviating catastrophic forgetting.

References

[1] Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2016. 2

[2] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. In *CVPR*, pages 303–312, 2021. 2

[3] You-Wei Luo and Chuan-Xian Ren. Mot: Masked optimal transport for partial domain adaptation. In *CVPR*, pages 3531–3540, 2023. 2